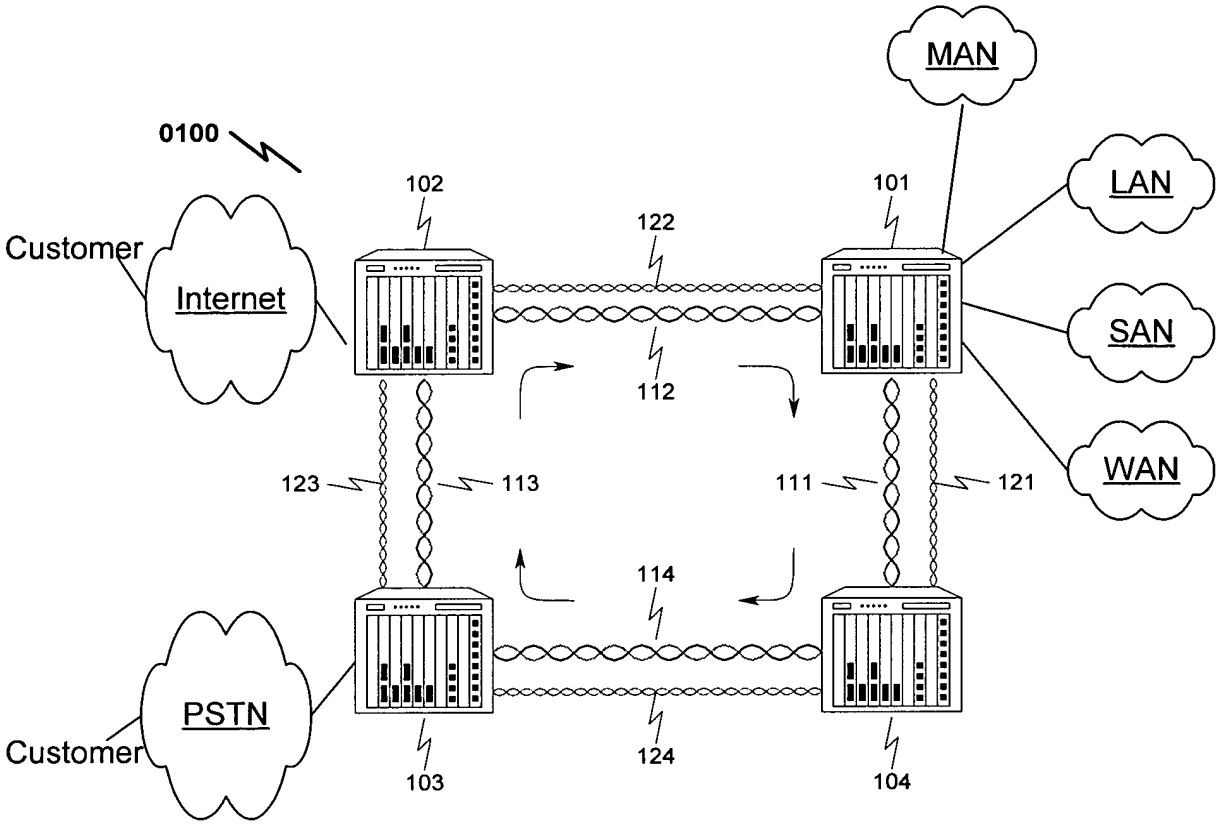
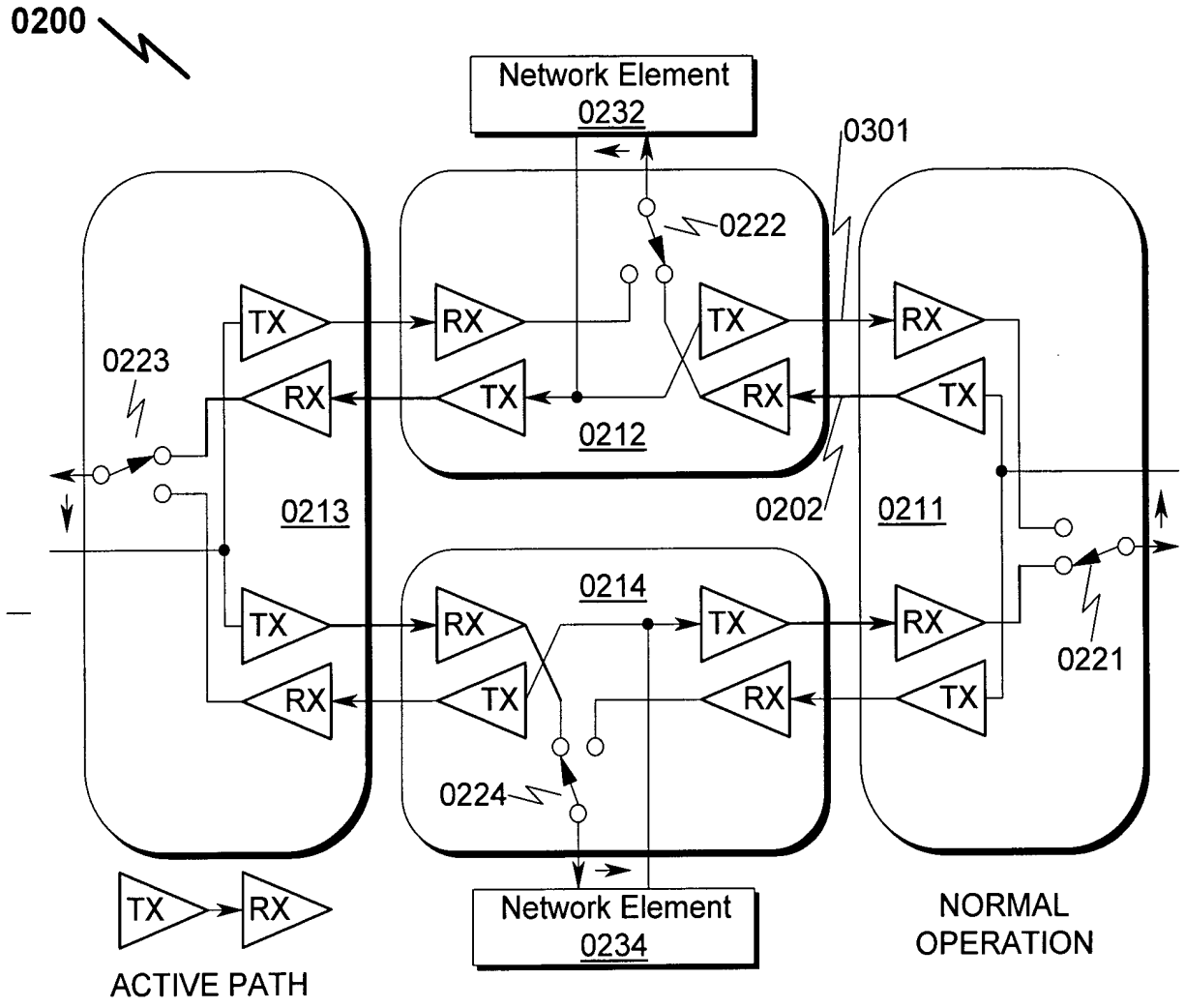


Figure 1



PRIOR ART

Figure 2

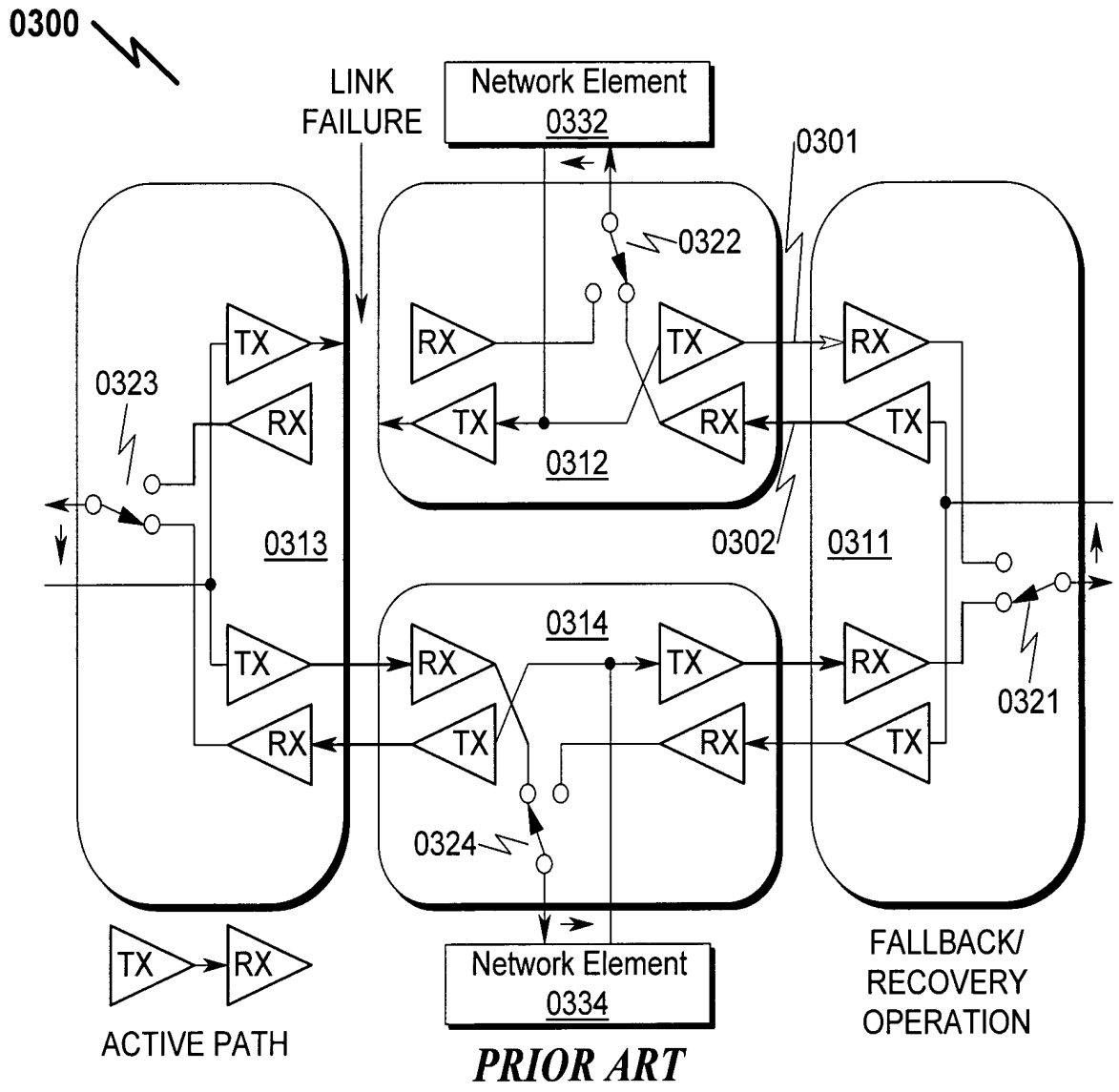
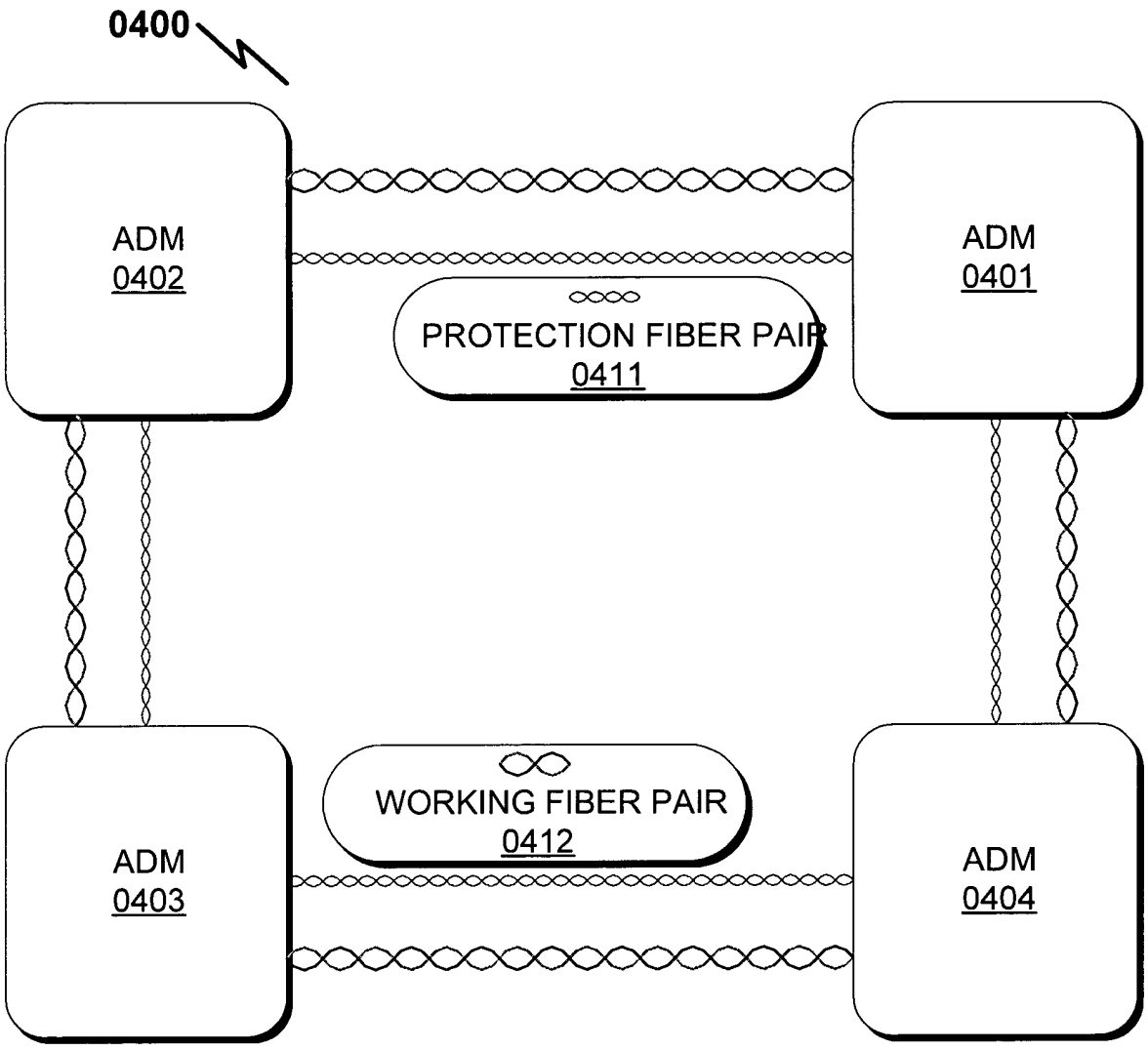

0300
LINK
FAILURE
Network Element
0332
0301
0322
TX
RX
0323
0312
0313
0302
0311
0314
0321
0324
Network Element
0334
FALLBACK/
RECOVERY
OPERATION
ACTIVE PATH
PRIOR ART

Figure 3



PRIOR ART

Figure 4

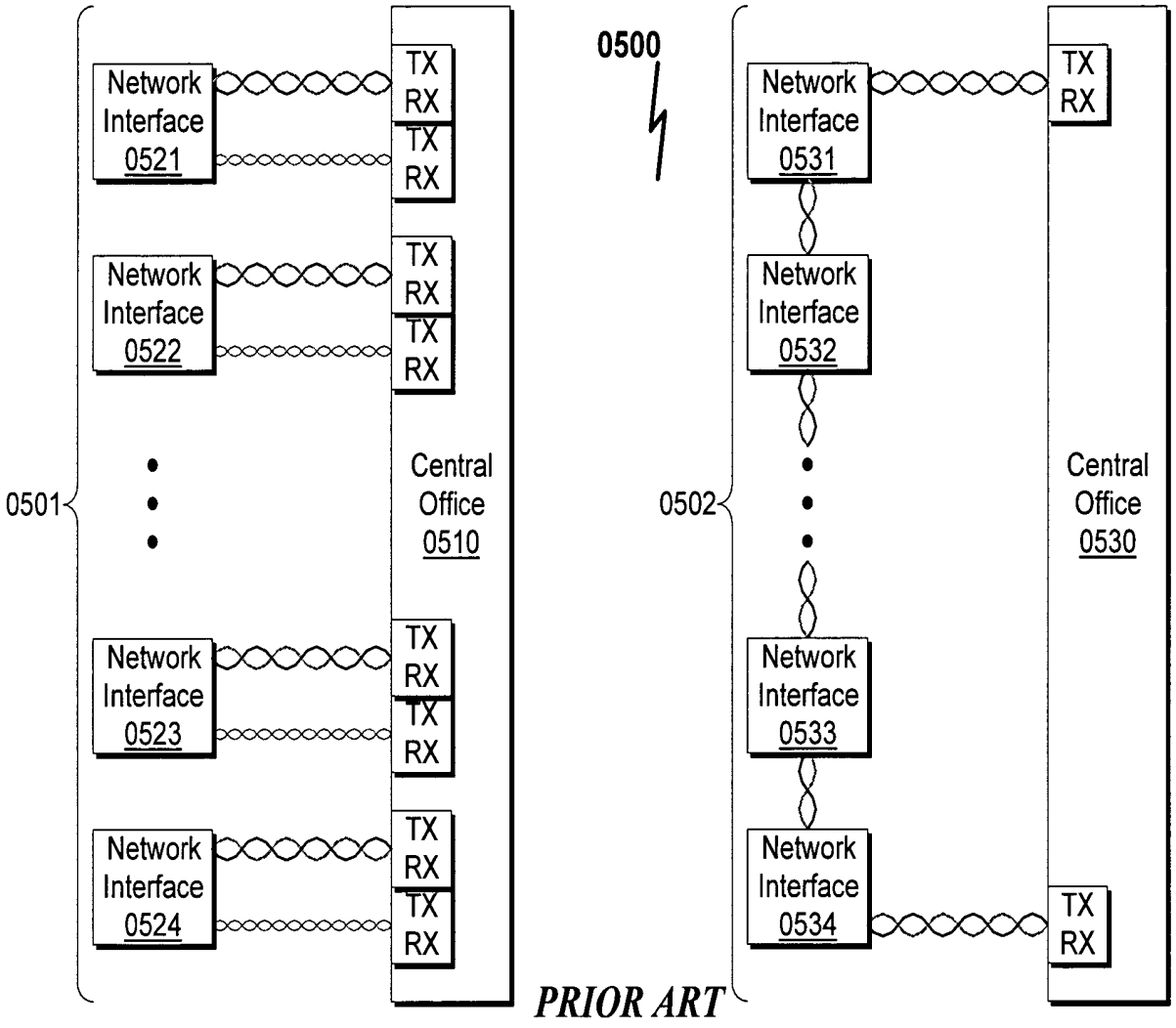

0500
Network Interface 0521
Network Interface 0522
Network Interface 0523
Network Interface 0524
TX
RX
0501
Central Office 0510
Network Interface 0531
Network Interface 0532
Network Interface 0533
Network Interface 0534
0502
Central Office 0530
PRIOR ART

Figure 5

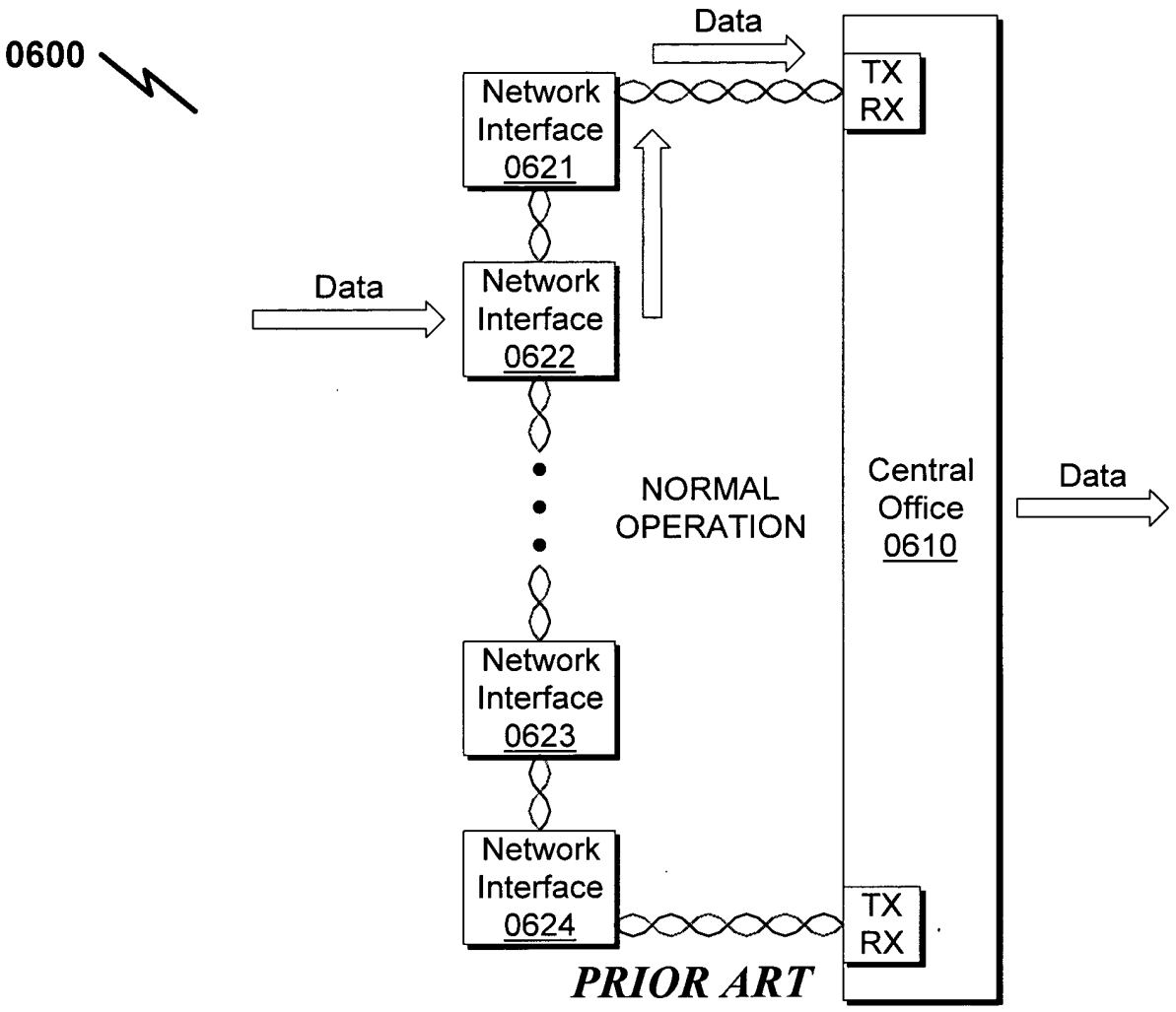

0600
Data
Network Interface 0621
TX RX
Data
Network Interface 0622
NORMAL OPERATION
Central Office 0610
Data
Network Interface 0623
Network Interface 0624
TX RX
PRIOR ART

Figure 6

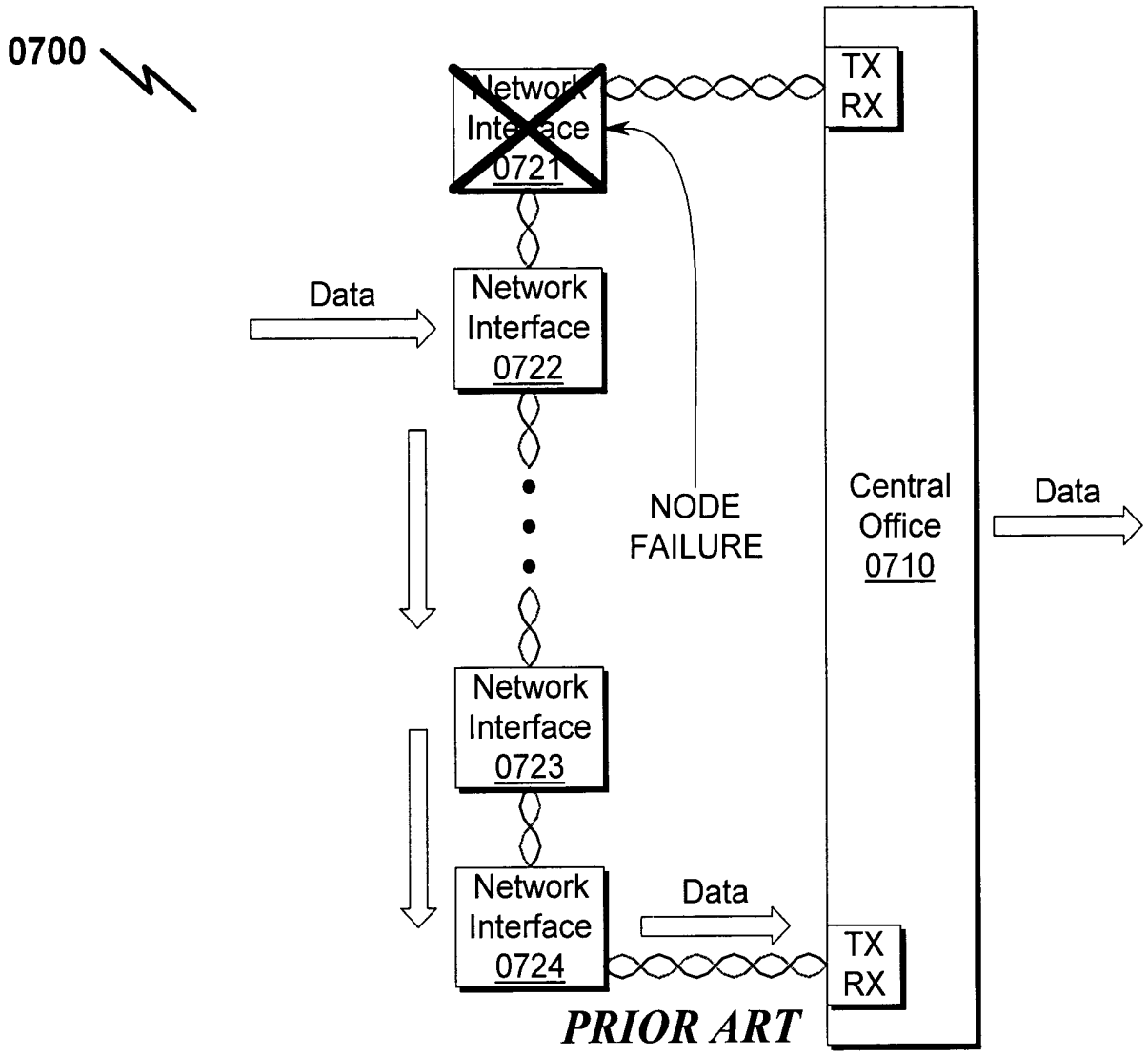

0700
Network Interface 0721
TX RX
Data
Network Interface 0722
NODE FAILURE
Central Office 0710
Data
Network Interface 0723
Network Interface 0724
Data
TX RX
PRIOR ART

Figure 7

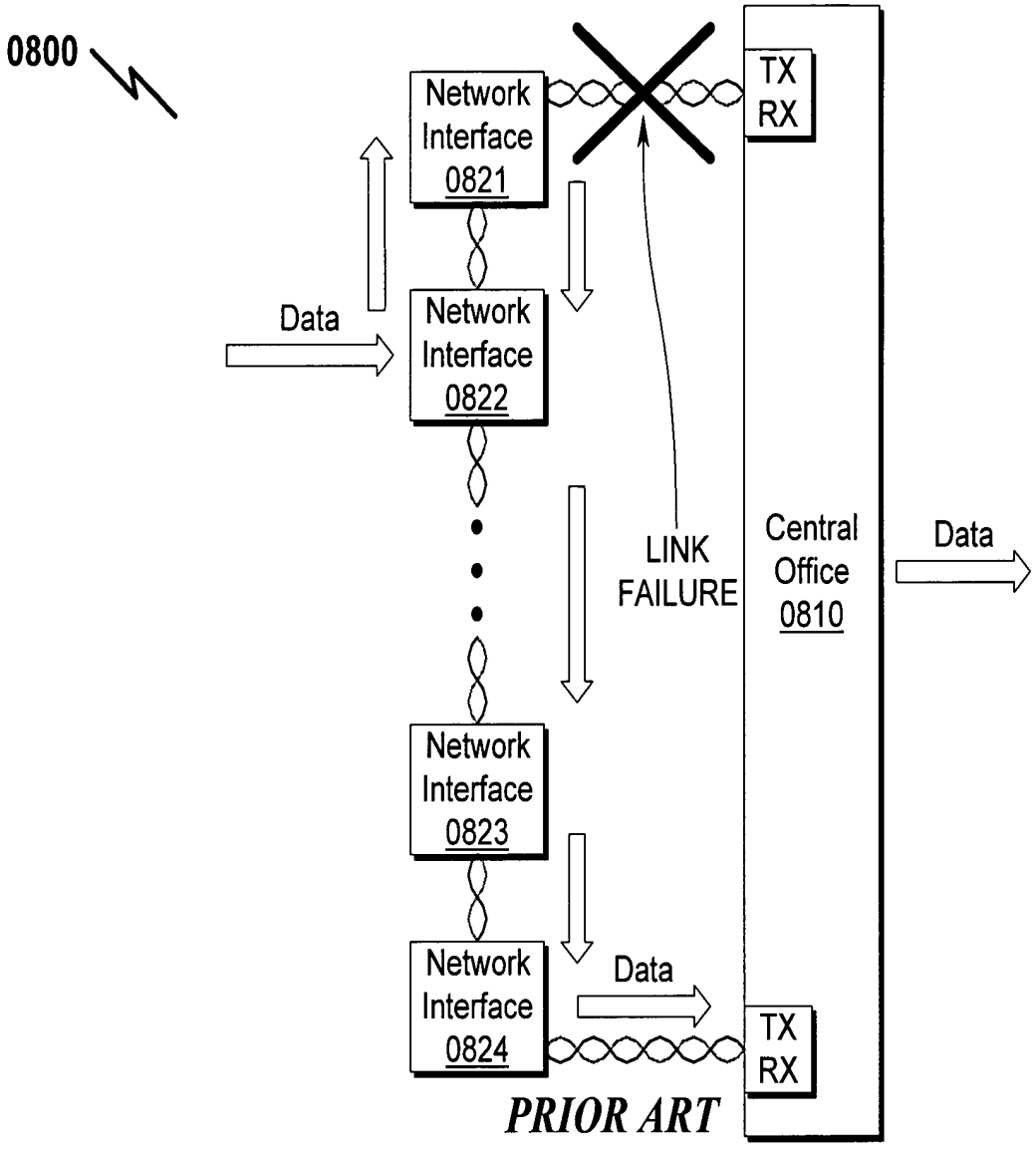

0800
Network Interface 0821
Network Interface 0822
Network Interface 0823
Network Interface 0824
TX RX
TX RX
Data
Data
Data
LINK FAILURE
Central Office 0810
PRIOR ART

Figure 8

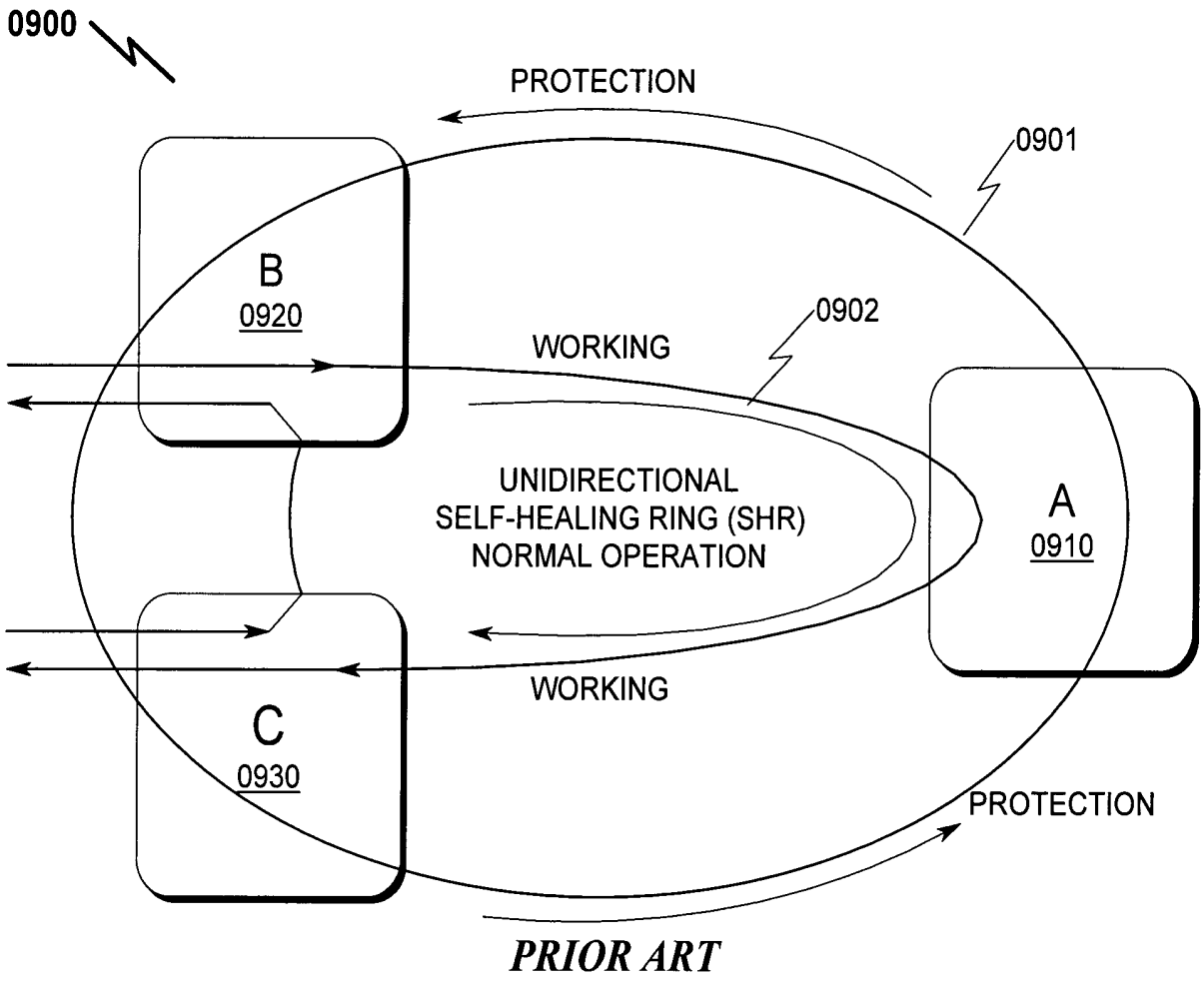

0900
PROTECTION
0901
B
0920
0902
WORKING
UNIDIRECTIONAL
SELF-HEALING RING (SHR)
NORMAL OPERATION
A
0910
WORKING
C
0930
PROTECTION
PRIOR ART

Figure 9

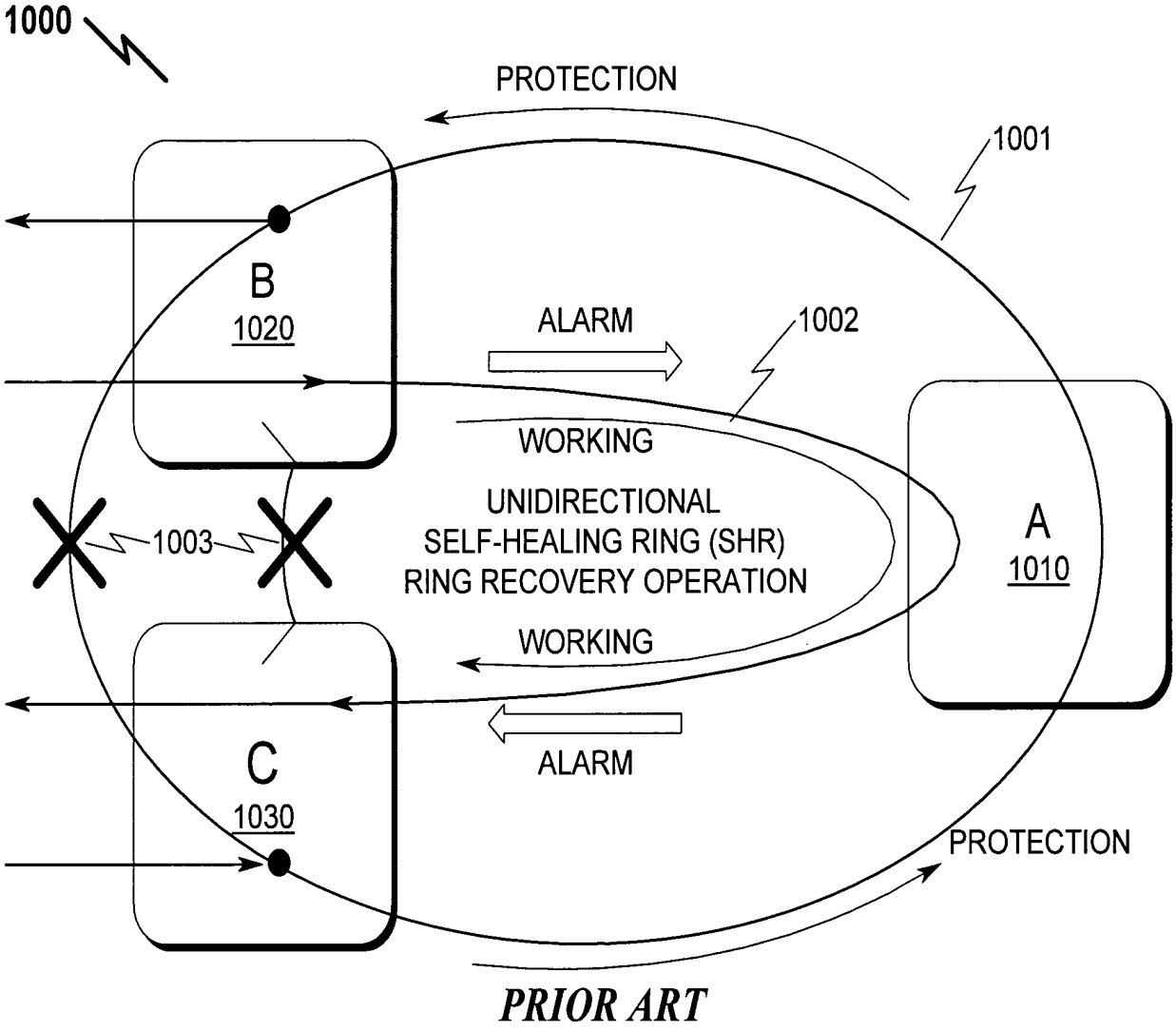

1000
PROTECTION
1001
B
1020
ALARM
1002
WORKING
UNIDIRECTIONAL
SELF-HEALING RING (SHR)
RING RECOVERY OPERATION
1003
A
1010
WORKING
ALARM
C
1030
PROTECTION
PRIOR ART

Figure 10

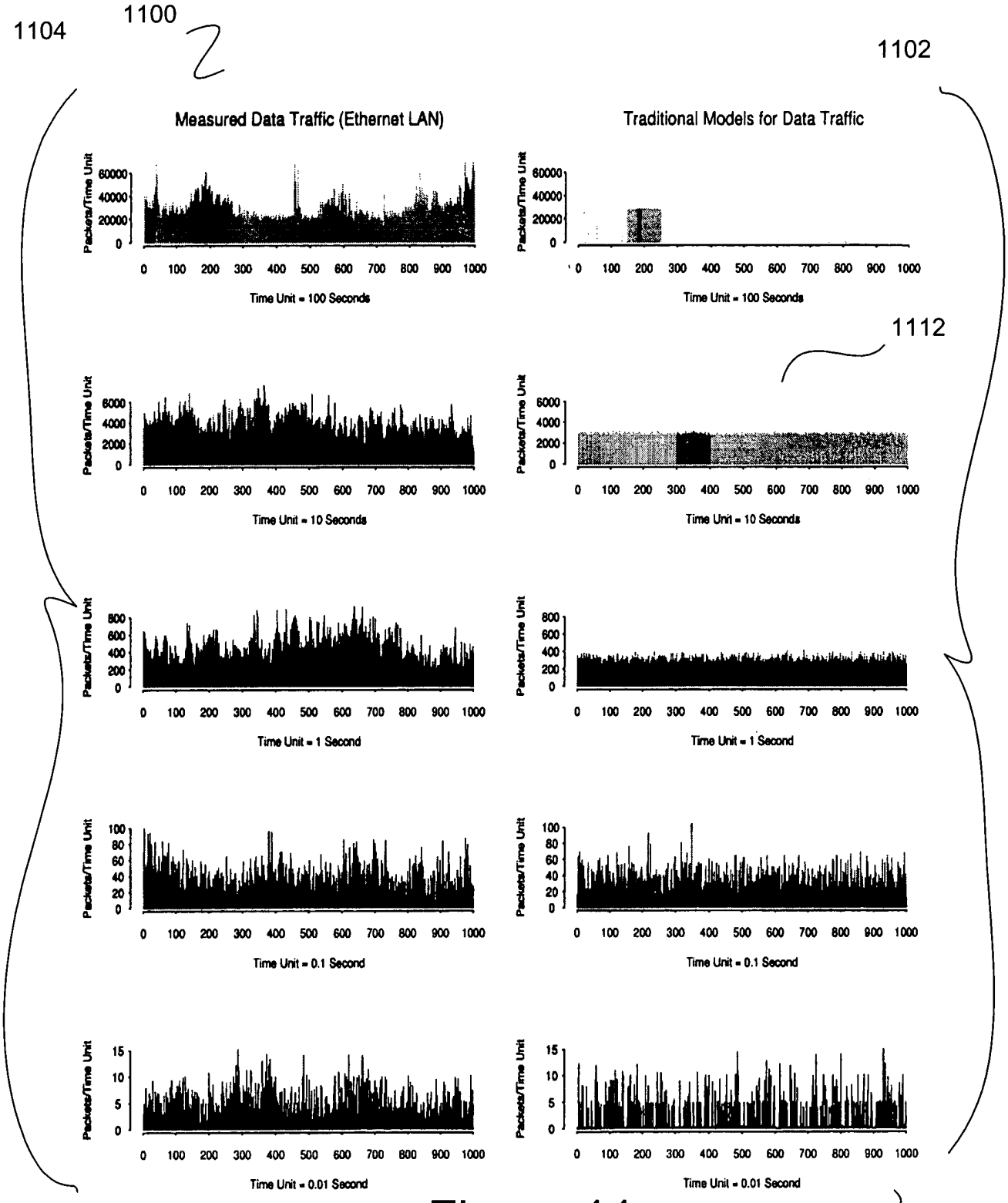


Figure 11

PRIOR ART

1200

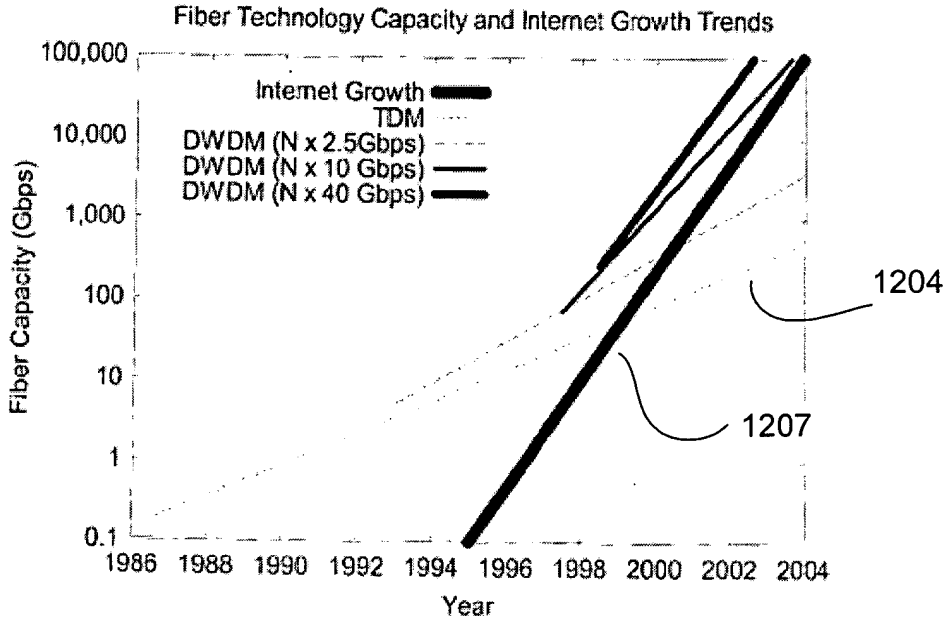


Figure 12

1300

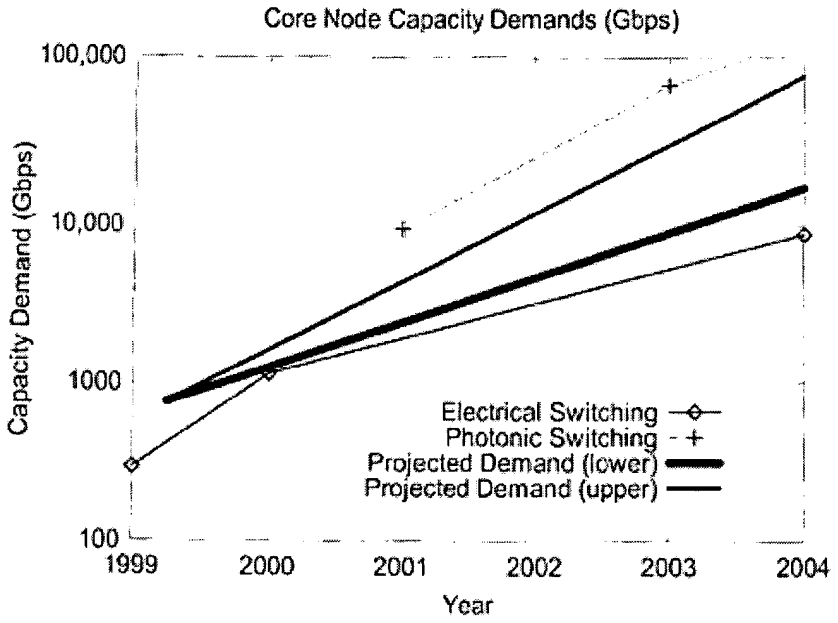


Figure 13

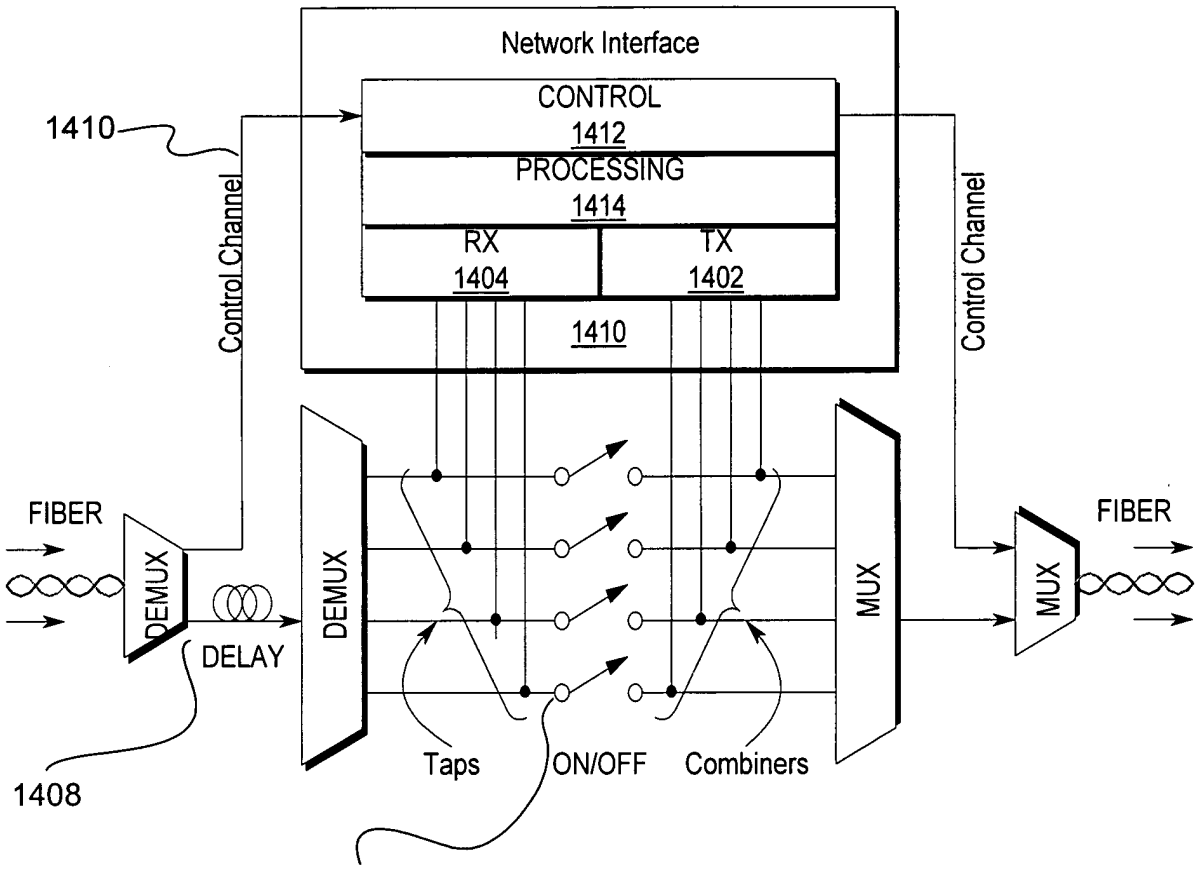


Figure 14

PRIOR ART

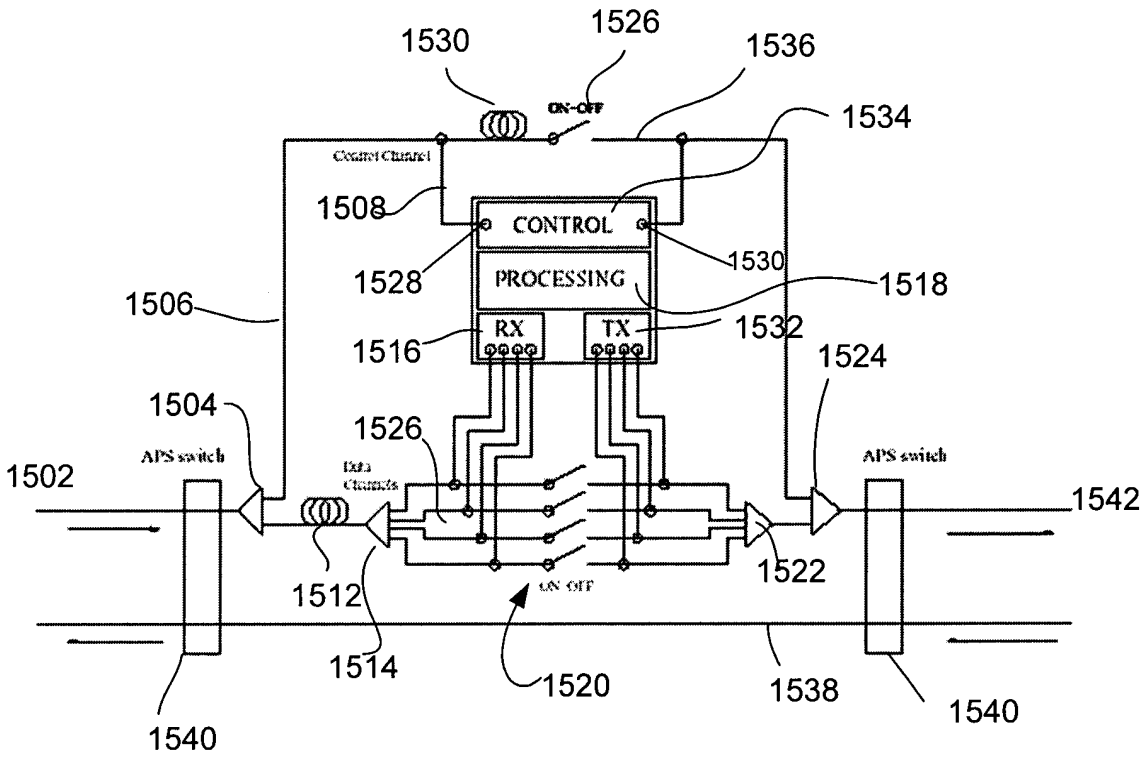

1530
1526
1536
ON-OFF
1534
Control Channel
1508
CONTROL
1528
1530
PROCESSING
1518
1506
RX
TX
1516
1532
1524
1504
1526
APS switch
APS switch
1502
Data Channels
1542
1512
1522
ON OFF
1514
1520
1538
1540
1540

Figure 15

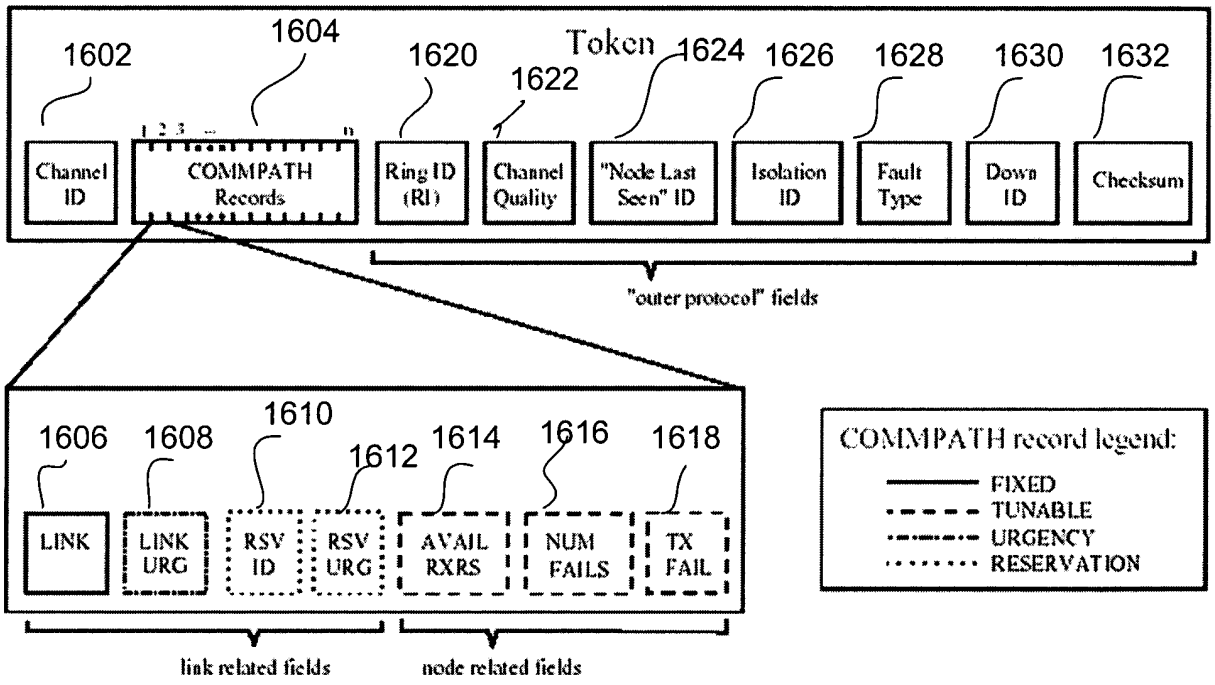
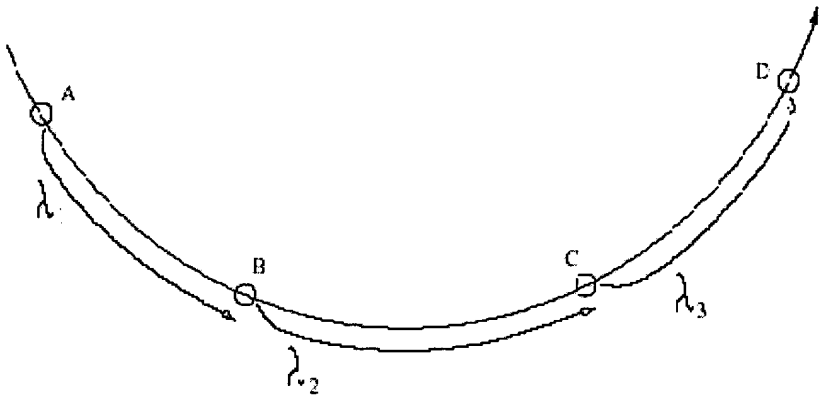
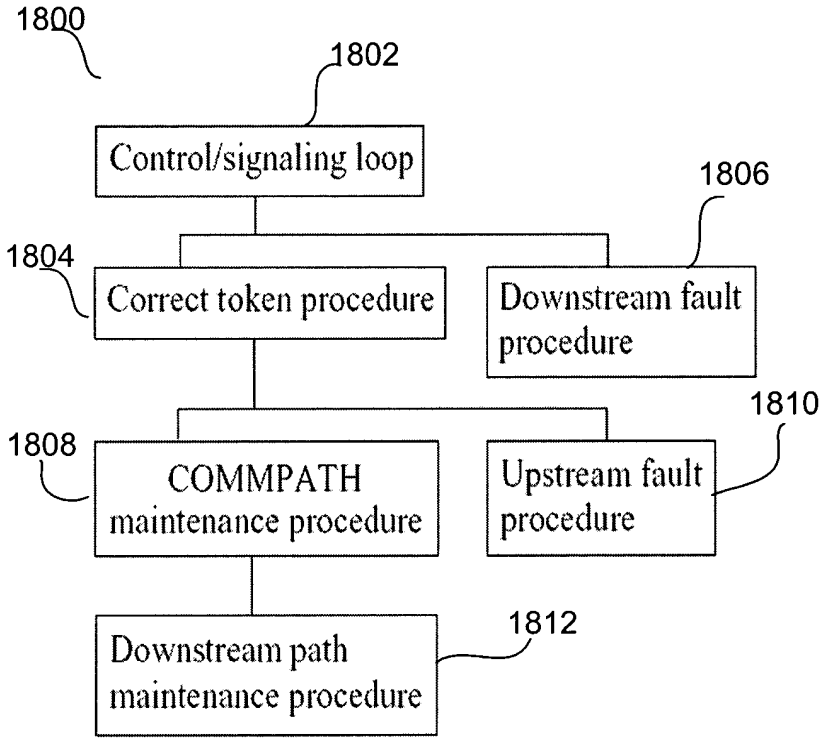

Token
1602
1604
1620
1622
1624
1626
1628
1630
1632
Channel ID
COMMPATH Records
Ring ID (RI)
Channel Quality
"Node Last Seen" ID
Isolation ID
Fault Type
Down ID
Checksum
"outer protocol" fields
1606
1608
1610
1612
1614
1616
1618
LINK
LINK URG
RSV ID
RSV URG
AVAIL RXRS
NUM FAILS
TX FAIL
link related fields
node related fields
COMMPATH record legend:
FIXED
TUNABLE
URGENCY
RESERVATION

Figure 16



Nodes A, B, and C contend for D's lone RXR

Figure 17



Reference Network protocol procedure dependencies

Figure 18

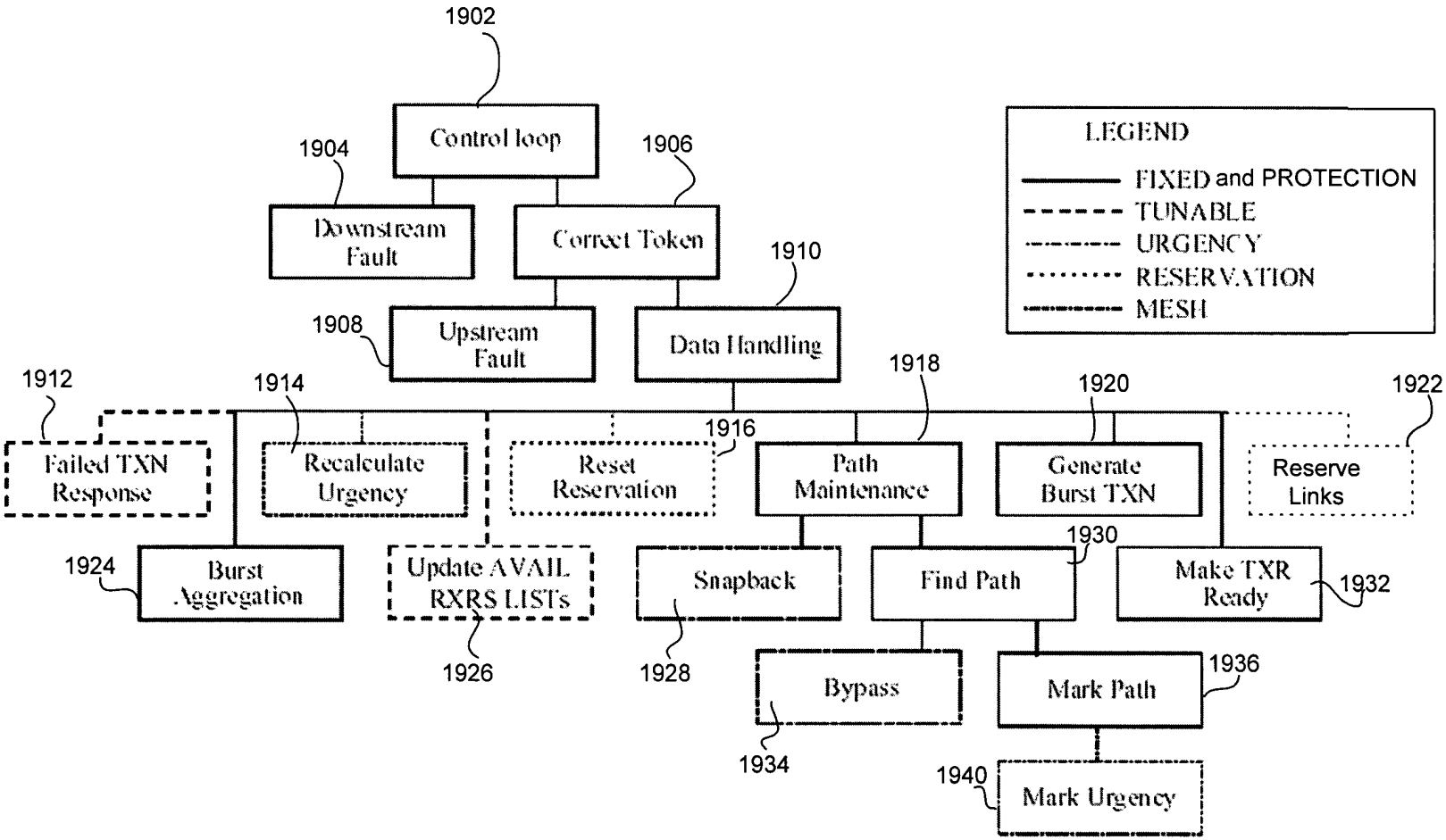

1902
Control loop
1904
Downstream Fault
1906
Correct Token
1908
Upstream Fault
1910
Data Handling
LEGEND
FIXED and PROTECTION
TUNABLE
URGENCY
RESERVATION
MESH
1912
Failed TXN Response
1914
Recalculate Urgency
1916
Reset Reservation
1918
Path Maintenance
1920
Generate Burst TXN
1922
Reserve Links
1924
Burst Aggregation
Update AVAIL RXRS LISTs
1926
1928
Snapback
1930
Find Path
Make TXR Ready
1932
1934
Bypass
1936
Mark Path
1940
Mark Urgency

Figure 19

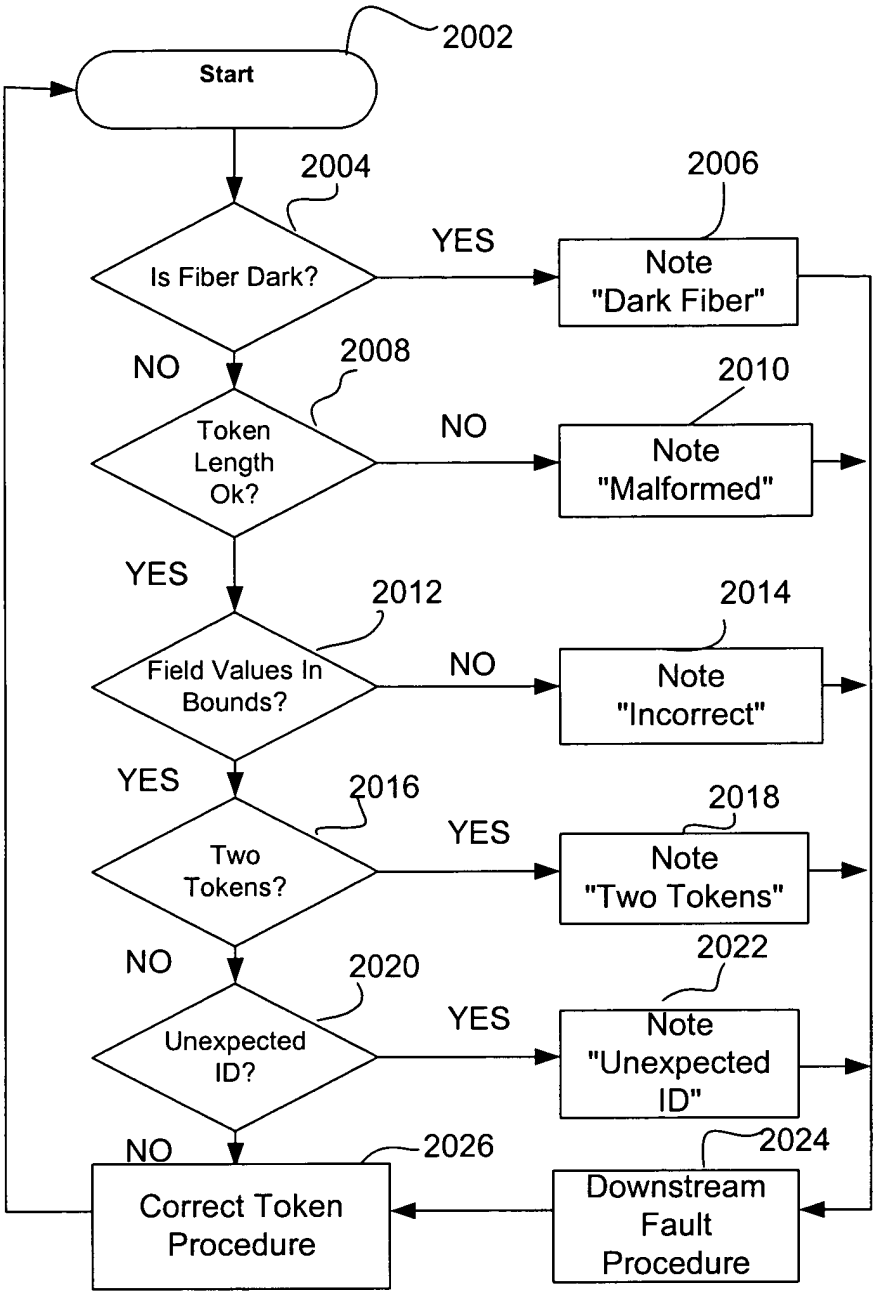

2002
Start
2004
Is Fiber Dark?
YES
2006
Note "Dark Fiber"
NO
2008
Token Length Ok?
NO
2010
Note "Malformed"
YES
2012
Field Values In Bounds?
NO
2014
Note "Incorrect"
YES
2016
Two Tokens?
YES
2018
Note "Two Tokens"
NO
2020
Unexpected ID?
YES
2022
Note "Unexpected ID"
NO
2026
Correct Token Procedure
2024
Downstream Fault Procedure

Figure 20

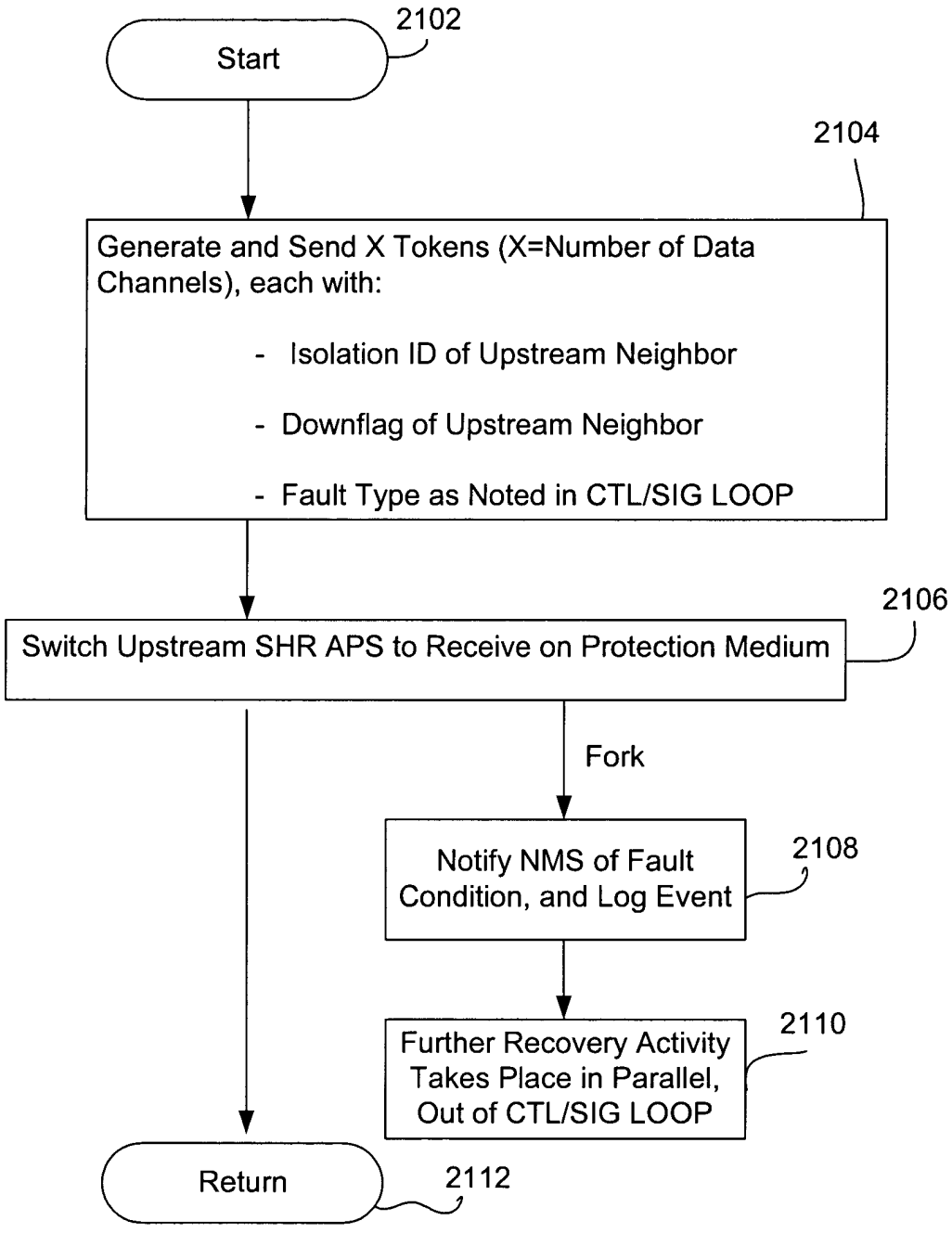

Start
2102
2104
Generate and Send X Tokens (X=Number of Data Channels), each with:
- Isolation ID of Upstream Neighbor
- Downflag of Upstream Neighbor
- Fault Type as Noted in CTL/SIG LOOP
2106
Switch Upstream SHR APS to Receive on Protection Medium
Fork
Notify NMS of Fault Condition, and Log Event
2108
Further Recovery Activity Takes Place in Parallel, Out of CTL/SIG LOOP
2110
Return
2112

Figure 21

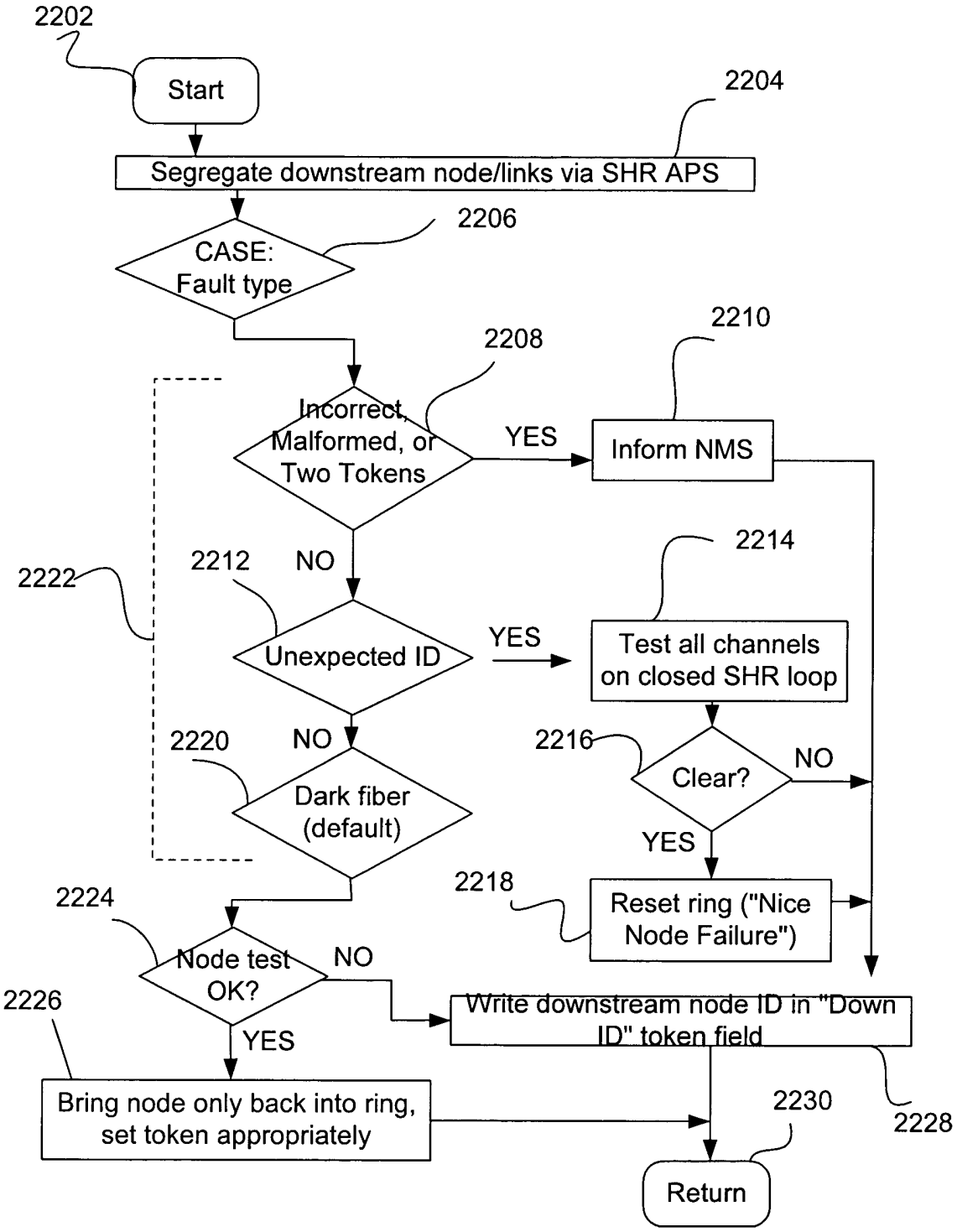

2202
Start
2204
Segregate downstream node/links via SHR APS
2206
CASE:
Fault type
2208
Incorrect, Malformed, or Two Tokens
YES
2210
Inform NMS
2212
NO
2214
2222
Unexpected ID
YES
Test all channels on closed SHR loop
2220
NO
2216
Clear?
NO
Dark fiber (default)
YES
2218
Reset ring ("Nice Node Failure")
2224
Node test OK?
NO
2226
Write downstream node ID in "Down ID" token field
YES
2228
Bring node only back into ring, set token appropriately
2230
Return

Figure 22

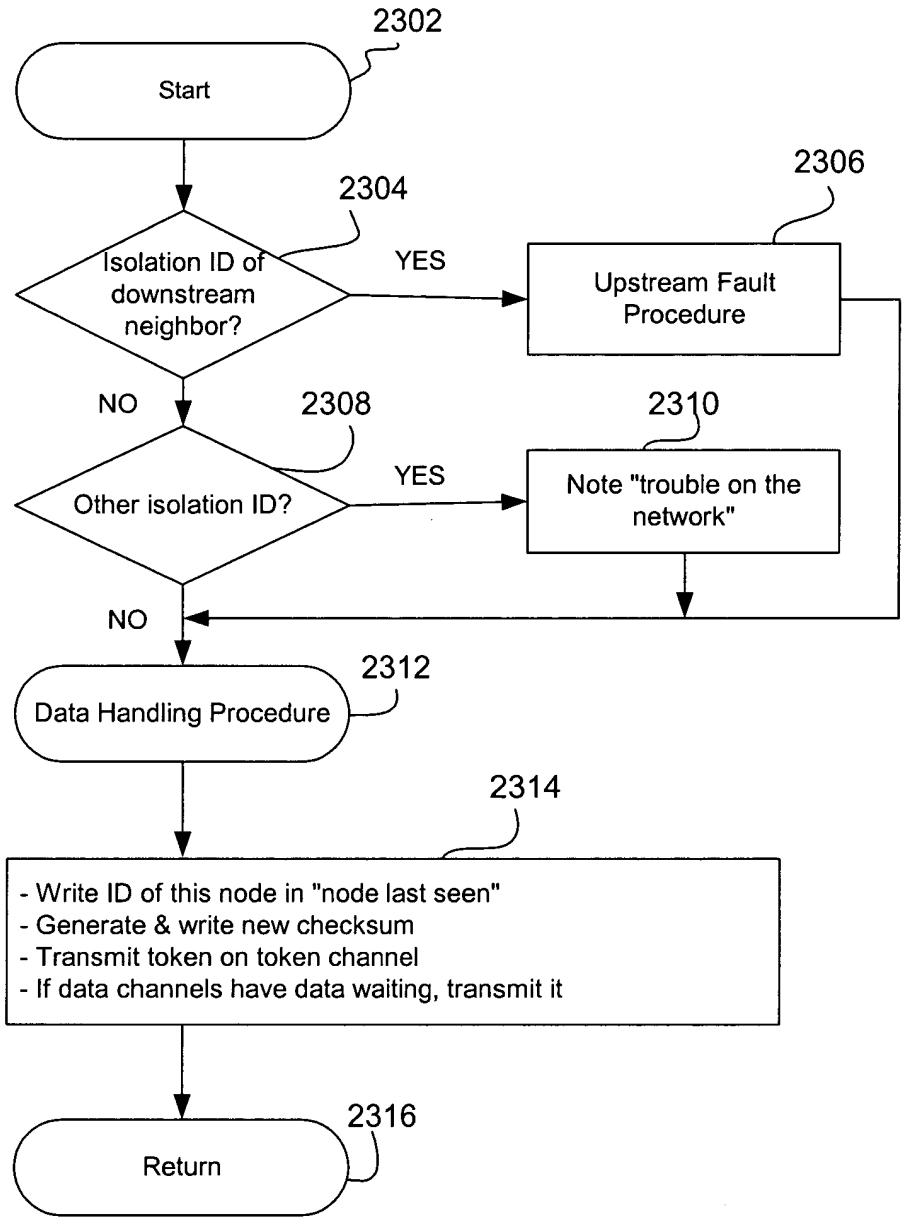

2302
Start
2304
Isolation ID of downstream neighbor?
YES
2306
Upstream Fault Procedure
NO
2308
Other isolation ID?
YES
2310
Note "trouble on the network"
NO
2312
Data Handling Procedure
2314
- Write ID of this node in "node last seen"
- Generate & write new checksum
- Transmit token on token channel
- If data channels have data waiting, transmit it
2316
Return

Figure 23

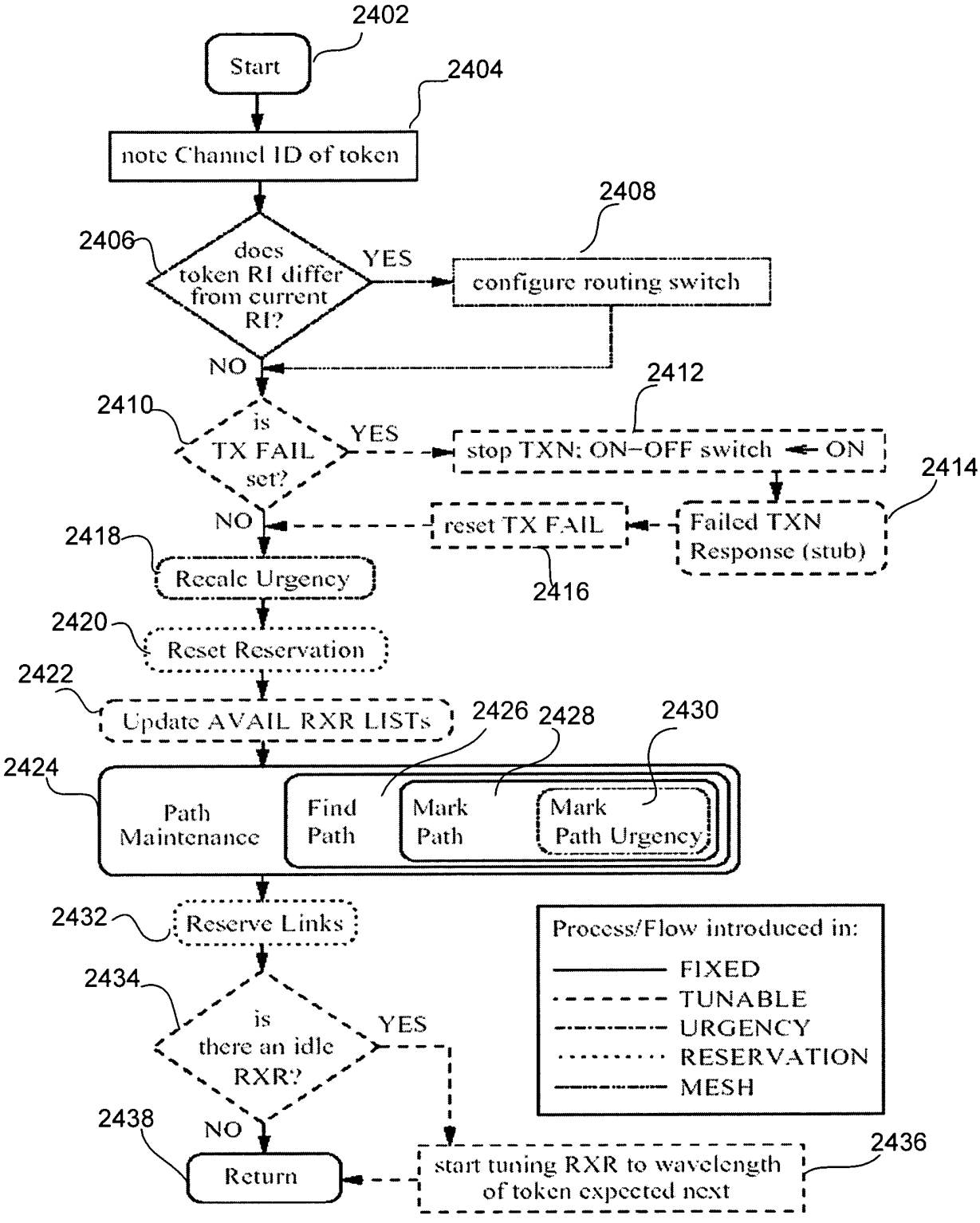

2402
Start
2404
note Channel ID of token
2408
2406
does token RI differ from current RI?
YES
configure routing switch
NO
2412
2410
is TX FAIL set?
YES
stop TXN; ON–OFF switch
ON
2414
NO
reset TX FAIL
Failed TXN Response (stub)
2418
Recalc Urgency
2416
2420
Reset Reservation
2422
Update AVAIL RXR LISTs
2426
2428
2430
2424
Path Maintenance
Find Path
Mark Path
Mark Path Urgency
2432
Reserve Links
Process/Flow introduced in:
FIXED
TUNABLE
URGENCY
RESERVATION
MESH
2434
is there an idle RXR?
YES
2438
NO
Return
start tuning RXR to wavelength of token expected next
2436

Figure 24

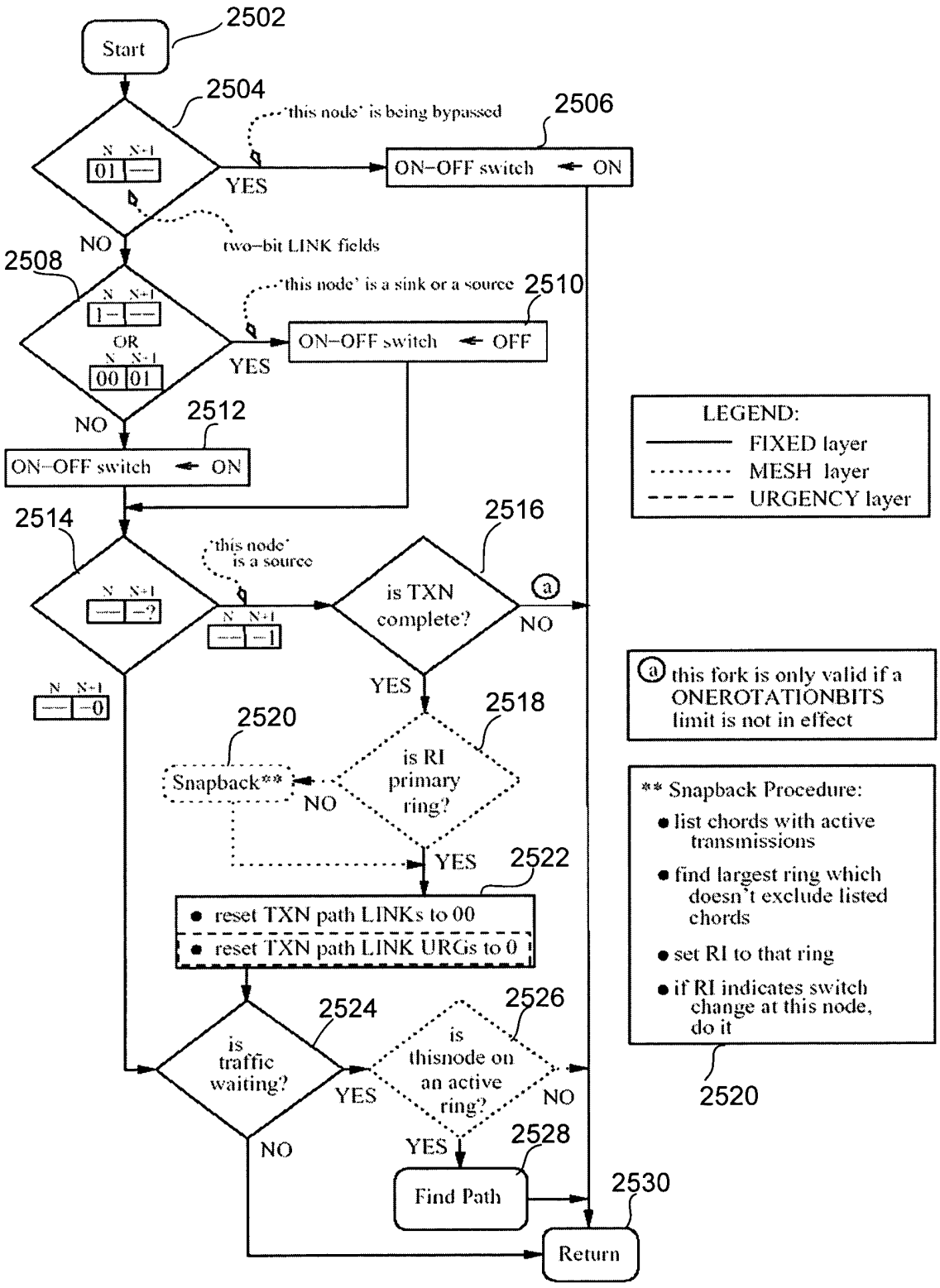
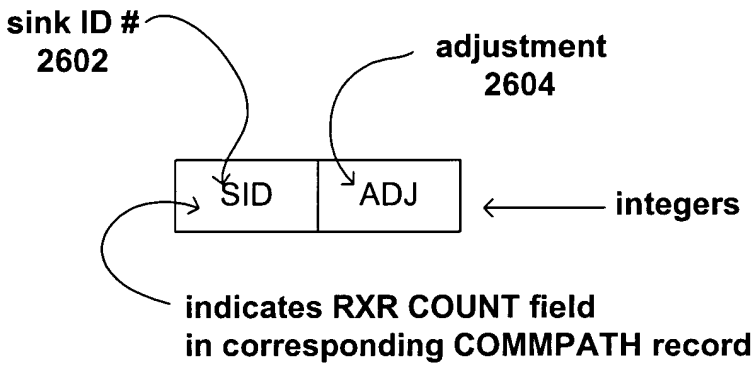

Start
2502
2504
'this node' is being bypassed
2506
N N+1
01 —
YES
ON–OFF switch ← ON
NO
two–bit LINK fields
2508
'this node' is a sink or a source
2510
N N+1
1– ––
OR
N N+1
00 01
YES
ON–OFF switch ← OFF
NO
2512
ON–OFF switch ← ON
2514
2516
'this node' is a source
N N+1
––
–?
N N+1
––
–1
is TXN complete?
a
NO
N N+1
––
–0
YES
2518
2520
is RI primary ring?
Snapback**
NO
YES
2522
• reset TXN path LINKs to 00
• reset TXN path LINK URGs to 0
2524
2526
is traffic waiting?
YES
is thisnode on an active ring?
NO
NO
YES
2528
Find Path
2530
Return
LEGEND:
FIXED layer
MESH layer
URGENCY layer
a this fork is only valid if a ONEROTATIONBITS limit is not in effect
** Snapback Procedure:
• list chords with active transmissions
• find largest ring which doesn't exclude listed chords
• set RI to that ring
• if RI indicates switch change at this node, do it
2520

Figure 25



RXR COUNT LIST record fields

Figure 26

Algorithm 0.0.2: UPDATE AVAIL RXR LISTs(*Global*, *Node*, *Token*)

```
if rxr_lists are empty                                                    [block 0]
  then return

if Node is on a lightpath                                                 [block 1]
  then note source and sink

for each rec ∈ Node.add_back_rxr_list                                     [block 2]
do { increment rec.adj
     increment Token[rec.sink].AVAIL_RXRS
     if Token[rec.sink].NUM_FAILS > 0
       then decrement Token[rec.sink].NUM_FAILS
     if rec.adj = 0
       then delete rec

for each rec_1 ∈ Node.take_away_rxr_list                                  [block 3]
do { decrement rec_1.adj
     decrement Token[rec_1.sink].AVAIL_RXRS
     if (Token[rec_1.sink].AVAIL_RXRS + Token[rec_1.sink].NUM_FAILS) < 0[a]
       then { increment Token[rec_1.sink].NUM_FAILS
              if sink noted and rec_1.sink = sink and Token[rec_1.sink].LINK = SINK[b]
                then if no active TXN to sink
                       then { comment: TANDEM
                              Node.on_off[λ_i] ← ON }
                else if Token[sink].LINK_URG ≥ urgency of least urgent active TXN
                       then { comment: STOMP
                              Node.on_off[λ_i] ← ON
                              discontinue own least urgent active TXN
                              invoke Failed_TXN() }
                       else { comment: SIPHON
                              reset lightpath from sink upstream
                              Node.on_off[λ_i] ← OFF
                              Token[source].TX_FAIL ← sink }
     if rec_1.adj = 0
       then { new rec_2 ← (rec_1.sink, −(Global.num_tokens))
              add rec_2 to Node.add_back_rxr_list
              delete rec_1 }
```

[a] if the sum of the AVAIL_RXRS and NUM_FAILS token fields for $rec_1.sink$ becomes negative ...

[b] if $rec_1.sink$ is the sink of a lightpath that was noted near the top of the algorithm ...

Figure 27

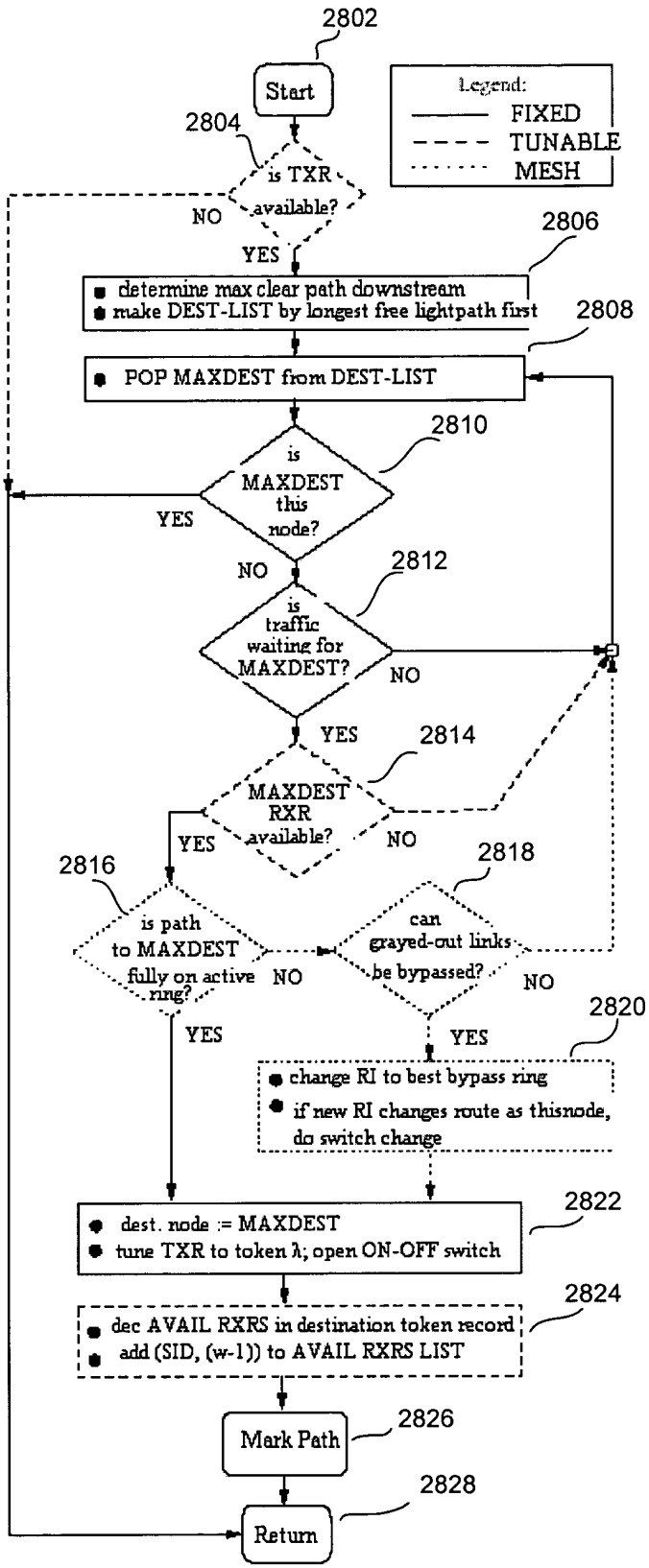

2802
Start
Legend:
FIXED
TUNABLE
MESH
2804
is TXR available?
NO
YES
2806
determine max clear path downstream
make DEST-LIST by longest free lightpath first
2808
POP MAXDEST from DEST-LIST
2810
is MAXDEST this node?
YES
NO
2812
is traffic waiting for MAXDEST?
NO
YES
2814
MAXDEST RXR available?
YES
NO
2816
is path to MAXDEST fully on active ring?
NO
YES
2818
can grayed-out links be bypassed?
NO
YES
2820
change RI to best bypass ring
if new RI changes route as thisnode, do switch change
2822
dest. node := MAXDEST
tune TXR to token λ; open ON-OFF switch
2824
dec AVAIL RXRS in destination token record
add (SID, (w-1)) to AVAIL RXRS LIST
Mark Path
2826
Return
2828

Figure 28

Algorithm 0.0.3: RESERVE LINKS PROCEDURE(*Token*, *Path*)

- create priority queue of destination *candidates*, sorted on primary key (most urgent), and secondary key (greatest hop-length)

while (1)

do
- **if** queue empty
  - **then return**
- pop *candidate*
- **if** every RSV_URG of path to *candidate* has lower urgency than *candidate* burst
  - **then** drop through WHILE loop

for each *link* on *Path*

do
- note "losing" reservation ID, if any
- write "my" ID and MAXDEST urgency number

for each losing ID

do
- reset all reservations which are contiguous to new reservation link and which have losing IDs (clear away "orphaned" IDs)

Figure 29

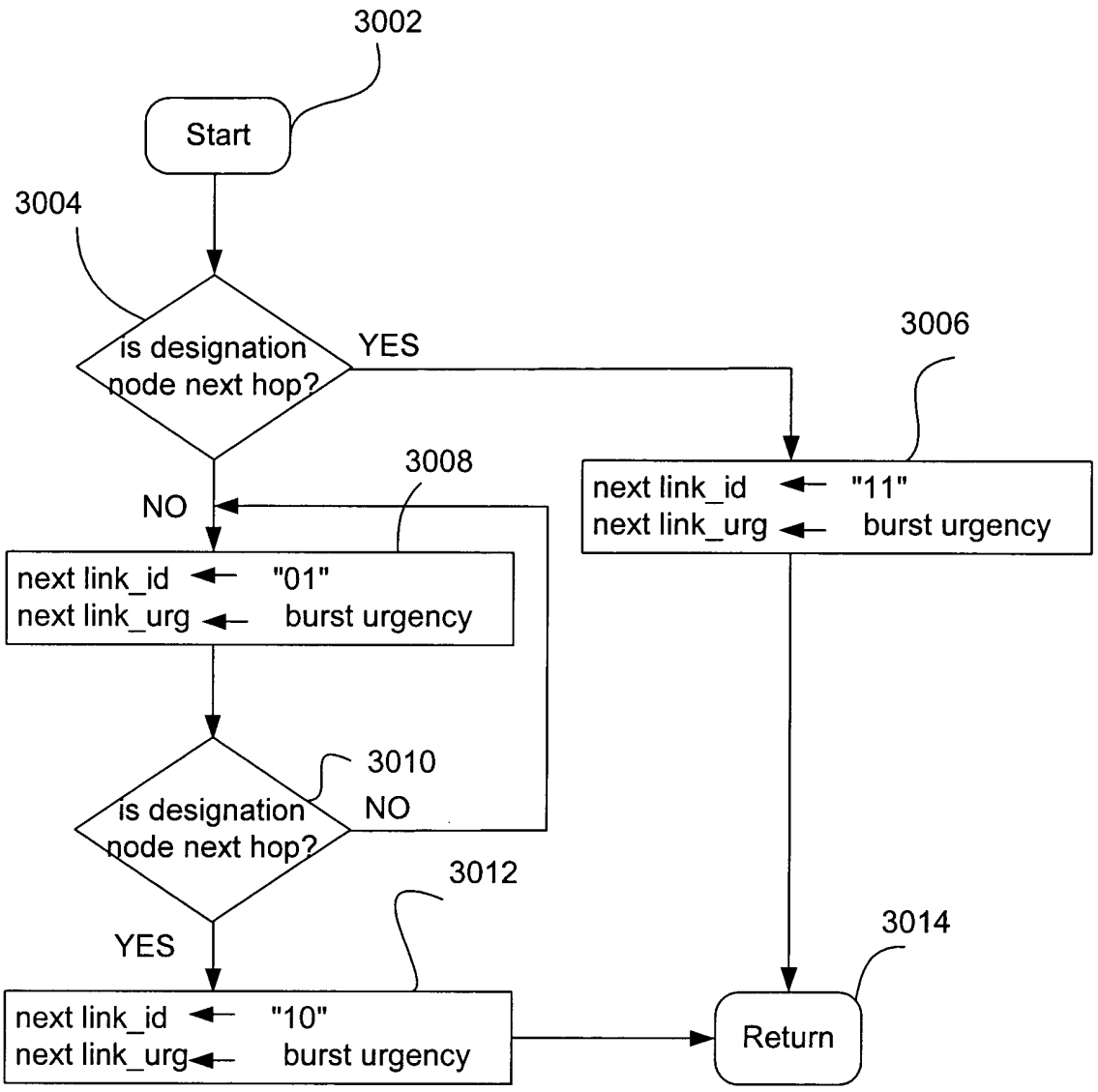

3002
Start
3004
is designation node next hop?
YES
3006
next link_id ← "11"
next link_urg ← burst urgency
NO
3008
next link_id ← "01"
next link_urg ← burst urgency
3010
is designation node next hop?
NO
3012
YES
next link_id ← "10"
next link_urg ← burst urgency
3014
Return

Figure 30

Algorithm 0.0.1: FIND PATH(*Global*, *Node*, *Token*)

```
if a TXR is available
then {
    find max (the FREE link farthest downstream on active ring)
    dest_list ← all dests (with bursts waiting) incl. max[a]
    sort dest_list, by primary key (most urgent)                    [3110]
                    and secondary key (farthest)
    while (1)
        do {
            if dest_list is empty
              then return
            dest ← pop dest_list
            if (∀ intermediate link, (dest.urg > link.RSV_URG)      [3118]
            and "grayed-out" links can be bypassed)                 [3120]
              then break[b]
        }
    if "grayed-out" links on path                                   [3130]
        then {
            Token.RI ← largest available bypass ring (RI)
            if new RI changes the route at thisnode, set switch
        }
    decrement Token[dest].AVAIL_RXRS
    arrec ← (dest, Global.num_tokens − 1)
    add arrec to Node.take_away_rxrs_list
    mark_path(dest, dest.urg)
}
```

[a] Recall that node information appears on the token in the same record with its upstream link.

[b] The break statement is unconditional, except in RESERVATION_SCHEME (first condition) and MESH (second condition).

Figure 31

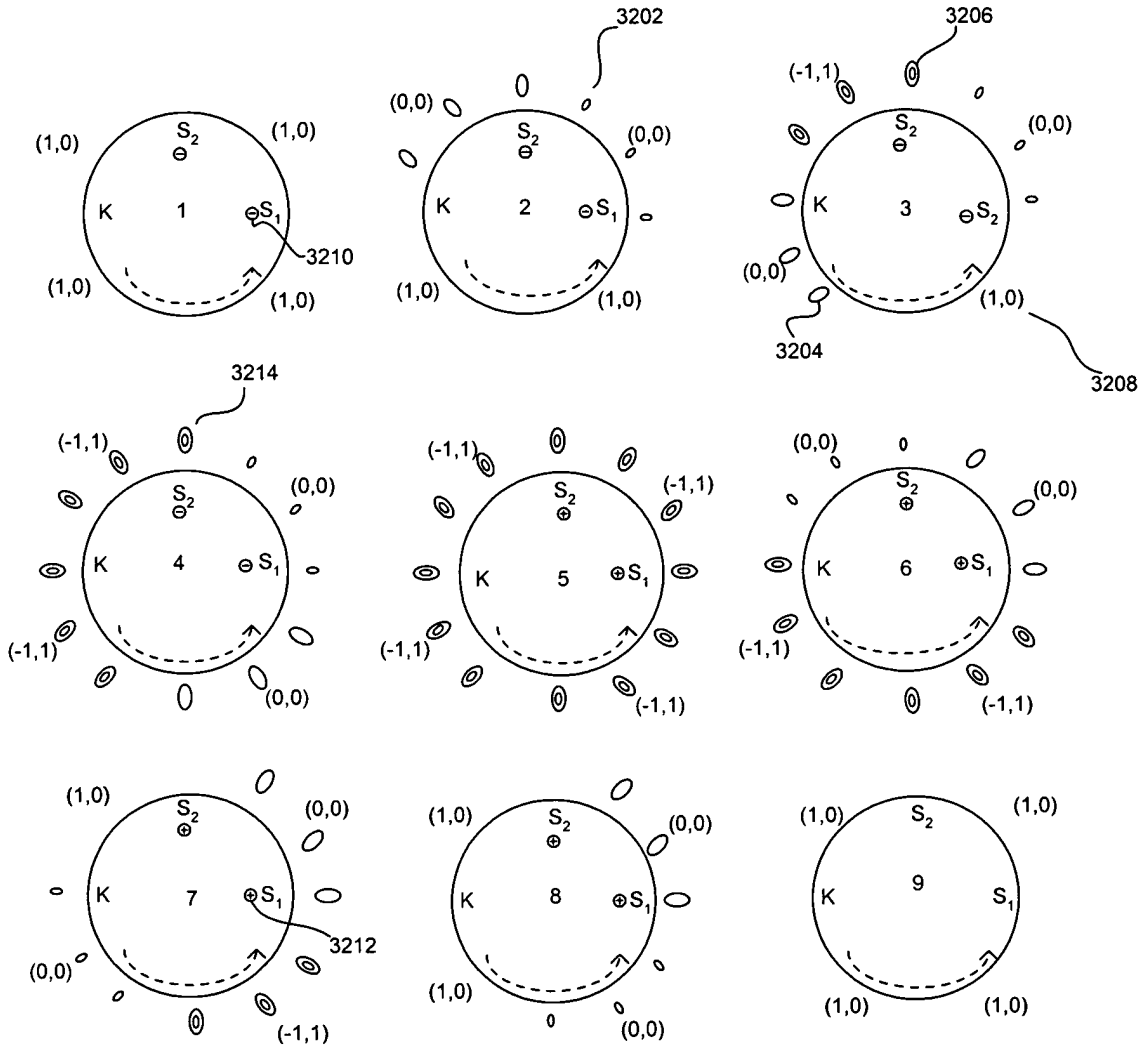
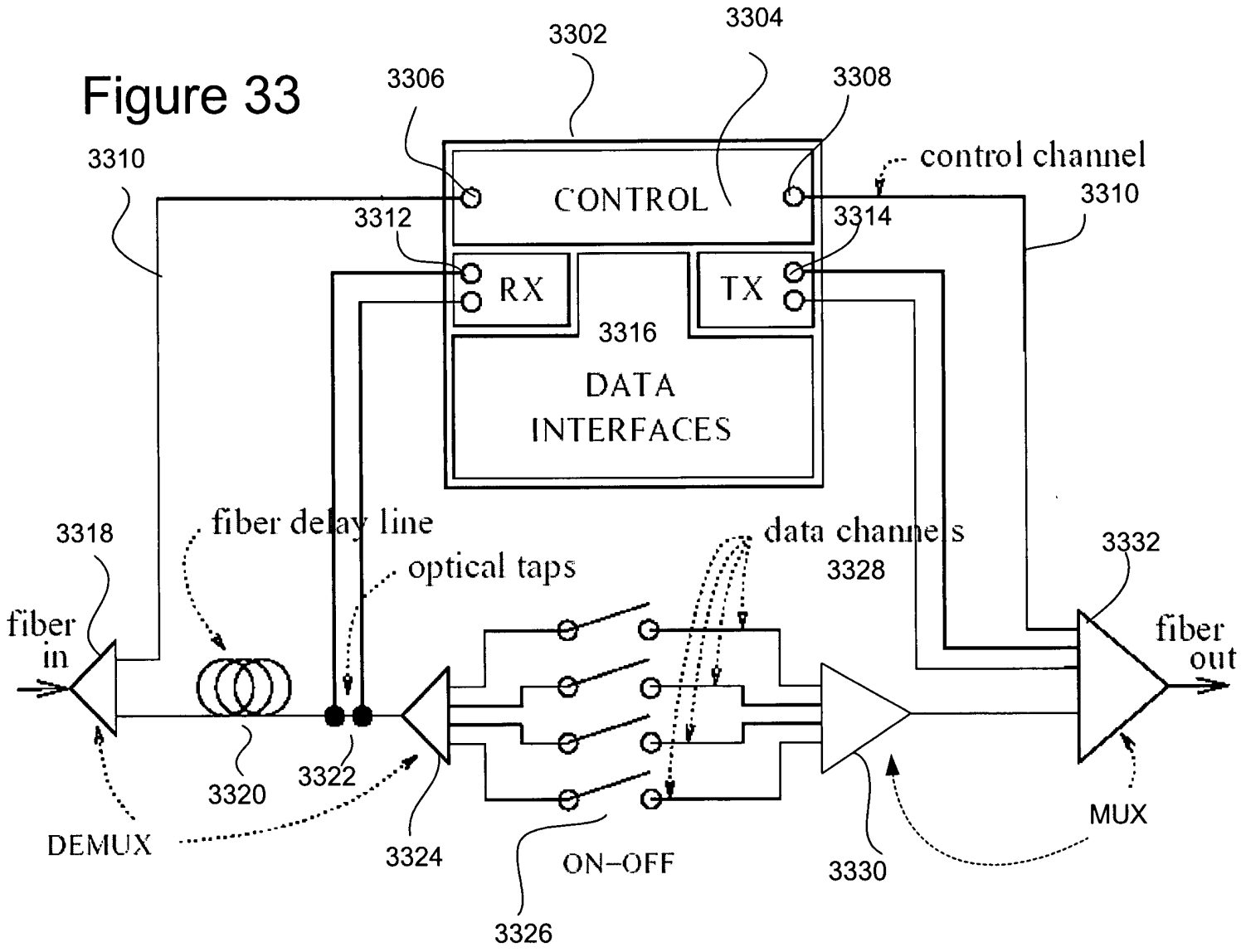


Figure 32

Figure 33


3302
3304
3306
3308
3310
CONTROL
control channel
3310
3312
3314
RX
TX
3316
DATA
INTERFACES
3318
fiber delay line
optical taps
data channels
3328
3332
fiber
in
fiber
out
3320
3322
DEMUX
3324
ON-OFF
3326
3330
MUX

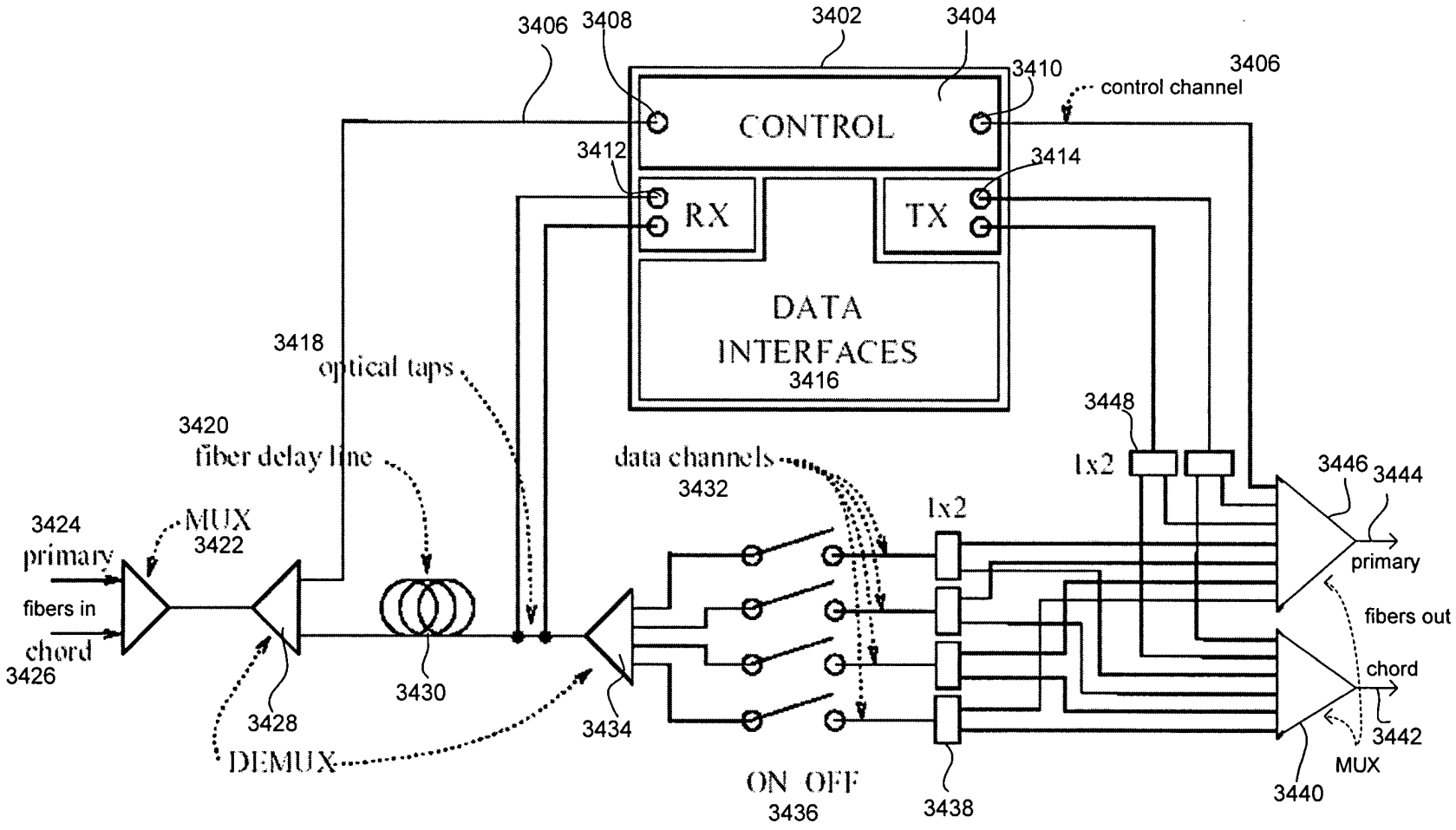

3406
3408
3402
3404
3410
3406
control channel
CONTROL
3412
3414
RX
TX
DATA
INTERFACES
3416
3418
optical taps
3420
fiber delay line
data channels
3432
3448
1x2
3446
3444
primary
3424
primary
MUX
3422
fibers in
chord
3426
1x2
fibers out
chord
3442
MUX
3440
3428
3430
3434
DEMUX
ON OFF
3436
3438

Figure 34